高等学校计算机应用规划教材

ASP动态网站开发教程（第三版）



- ASP基础与ASP开发环境的构建
- 网页设计基础
- ASP脚本语言VBScript语法基础
- ASP常用对象的功能及其使用
- ASP的常用组件及其使用
- ASP的文件处理
- ASP与Web数据库的连接和交互
- XML与ASP的应用
- ASP脚本调试和错误处理
- 动态网站开发综合实例

陈建伟　陈焕英　编著



清华大学出版社

高等学校计算机应用规划教材

ASP 动态网站开发教程

(第三版)

陈建伟　陈焕英　编著

清华大学出版社

北　京

内 容 简 介

本书由浅入深、循序渐进地介绍了使用 Microsoft 公司的 ASP 脚本开发 Web 应用程序的方法。全书共分 13 章。第 1 章介绍 ASP 的基础知识以及构建 ASP 开发环境的方法；第 2 章介绍设计网页的基础知识；第 3 章介绍 ASP 脚本语言 VBScript 语言的基本语法；第 4 章、第 5 章、第 7 章和第 10 章介绍 ASP 对象的基本知识和用法；第 6 章介绍 ASP 常用服务器组件的使用；第 8 和第 9 章介绍 ASP 与 Web 数据库之间的连接和交互方法；第 11 章介绍 XML 与 ASP 的应用；第 12 章介绍 ASP 脚本调试和自定义错误信息等；第 13 章通过 4 个综合实例对本书所介绍的知识进行了贯穿。

本书内容翔实，结构清晰，实例丰富，可操作性强，适合作为高等院校网站建设相关专业的教材，也可供 ASP 初学者和网站开发人员参考。

图书在版编目(CIP)数据

中国版本图书馆 CIP 数据核字(2007)第 159571 号

责任编辑：王定　陈小霞
装帧设计：
责任校对：胡雁翎
责任印制：
出版发行：清华大学出版社　　**地　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　编**：100084
c-service@tup.tsinghua.edu.cn
社 总 机：010-62770175　　**邮购热线**：010-62786544
投稿咨询：010-62772015　　**客户服务**：010-62776969
印 刷 者：
装 订 者：
经　销：全国新华书店
开　本：185×260　**印　张**：23　**字　数**：531 千字
版　次：2007 年 12 月第 1 版　**印　次**：2007 年 12 月第 1 次印刷
印　数：1~5000
定　价：33.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。
联系电话：010-62770177 转 3103　　产品编号：

前　　言

ASP 是 Microsoft 公司开发的服务器端的脚本环境，是目前流行的开放式 Web 应用程序开发技术。ASP 的主要特性是能够把脚本、HTML、组件和强大的 Web 数据库访问功能结合在一起，形成一个能在服务器上运行的应用程序，并把按用户要求专门制作的页面传给客户端浏览器。

ASP 属于 ActiveX 技术中的服务器端技术，与通常在客户端实现动态页面的技术(如 Java Applet、ActiveX 控件和 VBScript 等)不同，ASP 中的命令和脚本都是在服务器端解释执行，将执行后的结果生成 HTML 页面并发送给浏览器。由于脚本在服务器端执行，因而开发者不必担心浏览器是否能执行脚本。同时，由于只是将 HTML 页面发送到浏览器，在浏览器上看不到 ASP 程序，因此可以防止程序被窃取。另外，ASP 还具有许多显著的优点。例如，ASP 运行在 Web 服务器的同一进程中，能更快、更有效地处理客户请求；ASP 提供了更简单、更方便的数据库访问方法，使开发基于数据库驱动的 Web 应用程序更容易；ASP 支持 VBScript 和 JScript，并能以插件形式支持其他脚本语言，如 REXX 和 Perl 等。

本书共分 13 章。第 1 章介绍 ASP 的基础知识以及构建 ASP 开发环境的方法，第 2 章介绍网页设计的基础知识，第 3 章介绍 ASP 脚本语言 VBScript 语言的基本语法，第 4 章、第 5 章、第 7 章和第 10 章介绍 ASP 对象的基本知识和用法，第 6 章介绍 ASP 常用服务器组件的使用，第 8 和第 9 章介绍 ASP 与 Web 数据库之间的连接和交互方法，第 11 章介绍 XML 与 ASP 的应用，第 12 章介绍 ASP 脚本调试和自定义错误信息等，第 13 章通过 4 个综合实例对本书所介绍的知识进行了贯穿。

本书面向学习 ASP 和网站开发的初、中级用户，采用了由浅入深、循序渐进的讲述方法，在内容编写上充分考虑到初学者的实际需求，通过大量实用的操作指导和有代表性的实例，使读者直观、迅速地了解 ASP 的主要功能和动态网站的制作方法。另外，读者可以通过各章的习题来巩固书中所学的知识。

本书的电子课件和实例源程序可通过 http://www.tupwk.com.cn/downpage 下载，课后习题参考答案可发邮件至 wkservice@tup.tsinghua.edu.cn 获取。

本书是集体智慧的结晶，除封面署名的作者外，参与本书编写和制作的人员还有杜思明、高娟妮、方峻、严晓雯、张立浩、耿向华、王维、孔祥亮、牛静敏、张雪琴等。尽管我们在编写本书时已尽了最大努力，但由于各种条件的限制，加之作者水平有限，书中不足之处在所难免，希望读者批评指正。

编　　者

目　　录

第1章　导　　论

随着 Interne 和网络技术的日益发展，ASP 逐渐被人们接受并广泛应用。ASP 是一个基于 Web 服务器的开发环境，用户利用它可以方便地创建和执行动态、互动且高性能的 Web 服务器应用程序。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 了解 ASP 的技术特点
- 了解 ASP 的运行基础
- 了解 ASP 的开发环境
- 掌握安装与设置 IIS 的方法
- 掌握创建 ASP 应用程序的方法

1.1　ASP 与动态网页

ASP 内含于 IIS(Internet Information Server)中，是一种 Web 服务器端的开发环境。通过在普通 HTML 页面中嵌入的 ASP 脚本语言，可以产生和执行动态的、交互的、高性能的 Web 应用程序。ASP 采用脚本语言 VBScript(JScript)作为自己的开发语言。

1.1.1　动态网页

静态网页是不包含程序代码的网页，不会在服务器端执行。静态网页内容通常以 HTML 语言编写，在服务器端以.htm 或.html 文件格式存储。对于静态网页，服务器不执行任何程序就把 HTML 页面文件传给客户端的浏览器直接进行解读工作，所以网页的内容不会因为执行程序而出现不同的内容。

动态网页是指网页内含有程序代码，并会被服务器执行的网页。用户浏览动态网页须由服务器先执行网页中的程序，再将执行完的结果传送到用户浏览器中。动态网页和静态网页的区别在于，动态网页会在服务器中执行一些程序。由于执行程序时的条件不同，所以执行的结果也可能会有所不同，最终用户所看到的网页内容也将不同，所以称为动态网页。

1. 动态网页的页面特点

动态网页发布技术的出现使得网页从单纯的展示平台变成了网络交互平台，能够提供

如下所示的网页动态效果。

- 在网页中添加一个滚动显示的广告栏。
- 从 HTML 的表单中接收信息并且存储到数据库中。
- 根据不同访问者显示不同内容，创建个性化主页。
- 在主页中添加计数器。
- 根据用户浏览器的版本、类型和能力显示不同档次的内容。
- 跟踪用户网站上的活动信息并且存入日志文件。

2. 动态网页的开发技术

目前动态网页开发的 3 种主流技术是 ASP、PHP 和 JSP，这三者各有所长。它们都需要把脚本语言嵌入到 HTML 文档中。这三者的不同之处在于，ASP 学习简单、使用方便；PHP 软件免费，运行成本低；JSP 多平台支持，转换方便。

- ASP：ASP 主要为 HTML 编写人员提供了在服务器端运行脚本的环境，使 HTML 编写人员可以利用 VBScript 和 JScript 或其他第三方脚本语言来创建 ASP，实现有动态内容的网页，如计数器等。
- PHP：PHP 是一种跨平台的服务器端的嵌入式脚本语言，是技术人员在制作个人主页的过程中开发的小应用程序，并经过整理和进一步开发而形成的语言。它能使用户独自在多种操作系统下迅速地完成一个简单的 Web 应用程序。PHP 支持目前绝大多数数据库，并且是完全免费的，可以从 PHP 官方站点(http://www.php.net)上自由下载。用户可以不受限制地获得源码，甚至可以在其中加进自己需要的特色。
- JSP：JSP 的全称是 Java Server Pages，是由 Sun 公司提出，多家公司合作建立的一种动态网页技术。JSP 的突出特点是其开放的、跨平台的结构可以运行在几乎所有的服务器系统上。JSP 将 Java 程序段和 JSP 标记嵌入普通的 HTML 文档中。当客户端访问一个 JSP 网页时，就执行其中的程序段。Java 是一种成熟的跨平台的程序设计语言，可以实现丰富强大的功能。

ASP、PHP 和 JSP 语言都是面向 Web 服务器的技术，客户端浏览器不需要任何附加的软件支持，它们都提供在 HTML 代码中混合某种程序代码，由语言引擎解释执行程序代码的能力。

在 ASP、PHP 和 JSP 环境下，HTML 代码主要负责描述信息的显示样式，而程序代码则用来描述处理逻辑。普通的 HTML 页面只依赖于 Web 服务器，而 ASP、PHP 和 JSP 页面需要附加语言引擎分析和执行程序代码。程序代码的执行结果被重新嵌入到 HTML 代码中，然后一起发送给浏览器。

1.1.2　ASP

ASP(Active Server Pages)是一套由 Microsoft 公司开发的服务器端脚本执行环境。ASP 内含于 IIS 3.0 以上版本中，利用 ASP 用户可以结合 HTML 网页、ActiveX 组件和 ASP 指

令建立动态、交互的 Web 应用程序。本节作为全书的开端将通过简单的描述，重点介绍 ASP 的技术特点和运行基础，为下面的学习提供必要的理论支持。

1. ASP 的技术特点

ASP 技术是一种开放的，无需编译的应用环境，开发者可以用 HTML、Script 以及可以被重复使用的 ASP 组件创建出动态的、功能强大的并且基于 Web 的网页解决方案。总结其技术特点有以下几个方面。

- 可以使用 VBScript 和 JavaScript 等脚本语言，并同时结合 HTML 代码快速地制作出功能强大的网页应用程序。
- 可以与任何 ActiveX Scripting 语言兼容。
- 独立于浏览器，网站的访问者只需要在自己的客户端浏览器上执行 HTML 代码即可浏览 ASP 页面。
- 面向对象的设计，ASP 技术拥有良好的可扩充性。
- 良好的保密性，ASP 源程序不会因为客户端的访问而泄漏。
- 简单的编写环境，开发人员使用普通的文本编辑器即可进行 ASP 页面的设计与编写。

2. ASP 的技术功能

从应用的角度看，利用 ASP 工作时几乎不存在任何限制。用户可以参照如下的 ASP 技术的典型应用介绍来了解 ASP 技术的具体功能。

- 在网页中添加计数器、留言本、公告板、聊天室等功能，实现以前的 CGI(Common Gateway Interface)程序的所有功能。
- 根据客户端浏览器的版本、类型以及处理能力显示不同档次的页面内容。使用户可以最大程度地欣赏网站所提供的网页信息。
- 从客户端浏览器提交的 HTML 表单中接收信息并能够联系数据库系统，及时地做出响应。
- 在网页中添加滚动显示的广告栏或标题内容信息。
- 根据访问者的需求显示不同的页面内容。
- 实现网站内多个页面之间的信息共享。
- 读写服务器文件，实现网站主页访问人次计数等功能。
- 跟踪用户网站中的活动信息并存入日志文件，记录访问者的访问行为。

3. ASP 的运行环境

ASP 常用的运行环境是 Microsoft Internet Information Server version 3.0/4.0 on Windows NT Server(Windows NT 服务器)和 Microsoft Personal Web Server on Windows 95/98(Windows 95/98 环境)。也就是说，在 Windows 95/98/2000/XP 环境中都可以使用 ASP。

另外，在 Windows 95/98 环境中使用 ASP 需要用户手动安装 PWS(Microsoft Personal Web Server)，在 Windows 2000/XP 环境中则需要安装 IIS(Microsoft Internet Information

Server)。因为本书将以 Microsoft Internet Information Server version 3.0/4.0 on Windows NT Server 环境为基础介绍 ASP 的使用，在后面的章节中将会介绍 IIS 的安装与使用。

4. ASP 的适用环境

虽然 ASP 技术功能强大，在管理和维护方面有着显著的优势，但也并非适用于任何环境。例如，在 UNIX 主机上虽然可以使用 ASP，但是由于 ASP 是 Microsoft 公司基于 NT 环境所开发的技术，在 UNIX 系统中难免会出现问题。

另外，由于 ASP 可以实现以往 CGI 的所有功能，如果设计者对 CGI 语言(Perl 和 C 等)不是十分熟悉，可以考虑使用 ASP 技术。因为 ASP 所使用的 VBScript 相对比较容易。但是如果 CGI 程序需要在 UNIX 主机上运行，最好选择例如 Perl 这种同时支持 NT 和 UNIX 环境的编写语言，方便日后进行程序移植。

1.2　构建 ASP 开发环境

在进行 ASP 网页开发之前，首先必须在本地计算机上搭建编译 ASP 动态网页所需的软件环境，包括配置本地计算机 IP 地址、安装与设置 IIS，以及创建虚拟目录等一系列的操作。本书将以 Windows XP 操作系统自带的 IIS 为例，来讲述其安装和配置方法。

1.2.1　配置 Internet 服务器

在这一节中，将详细介绍如何配置一台具有 WWW 服务功能的 Internet 服务器的第一个步骤，即设置本地计算机的 IP 地址。

1. IP 地址简介

所谓 IP(Internet Protocol)地址实际上就是一种用于标记网络节点和指定路由选择信息的方式。一个 IP 地址被用来标识网络中的一个通信实体，例如一台主机，或是路由器的某一个端口。而在基于 IP 协议网络中传输的数据包，也都必须使用 IP 地址来进行标识，如同我们写一封信，要标明收信人的通信地址和发信人的地址，邮政工作人员则通过该地址来决定邮件的去向。因此对于联入网络的计算机而言，必须给它们分配唯一的 IP 地址以保证其在网络中的唯一性。

通常，IP 地址由网络标识符与网络管理员分配的唯一主机标识符组成。IP 地址的表示方法是带小数点的十进制记数法，其中每 8 位字节的十进制值用“.”号分隔，例如 192.168.0.1 或 167.0.0.11 等。

2. 配置 IP 地址

为本地计算机配置 IP 地址的前提是计算机上必须安装有网络适配器(网卡)。下面以

Windows XP 操作系统为例，介绍为网卡配置 IP 地址的基本方法。

【练习 1-1】在 Windows XP 操作系统中设置本地计算机的 IP 地址为 192.168.0.81，子网掩码为 255.255.255.0，默认网关为 192.168.0.1。

(1) 选择“开始”|“设置”|“控制面板”命令，在打开的“控制面板”窗口中双击“网络连接”图标，打开“网络连接”窗口。

(2) 在“网络连接”窗口中右击“本地连接”图标，在弹出的快捷菜单中选择“属性”命令，打开“本地连接属性”对话框。

(3) 选择“常规”选项卡，在“此连接使用下列项目”列表框中选择“Internet 协议(TCP/IP)”选项，如图 1-1 所示，然后单击“属性”按钮，打开“Internet 协议(TCP/IP)属性”对话框。

(4) 选中“使用下面的 IP 地址”单选按钮，然后在“IP 地址”文本框中输入为本地计算机设定的 IP 地址 192.168.0.81，在“子网掩码”文本框中输入子网掩码 255.255.255.0，在“默认网关”文本框中输入默认网关地址 192.168.0.1，如图 1-2 所示。

(5) 完成设置后，单击“确定”按钮保存设置，然后重新启动系统即可。

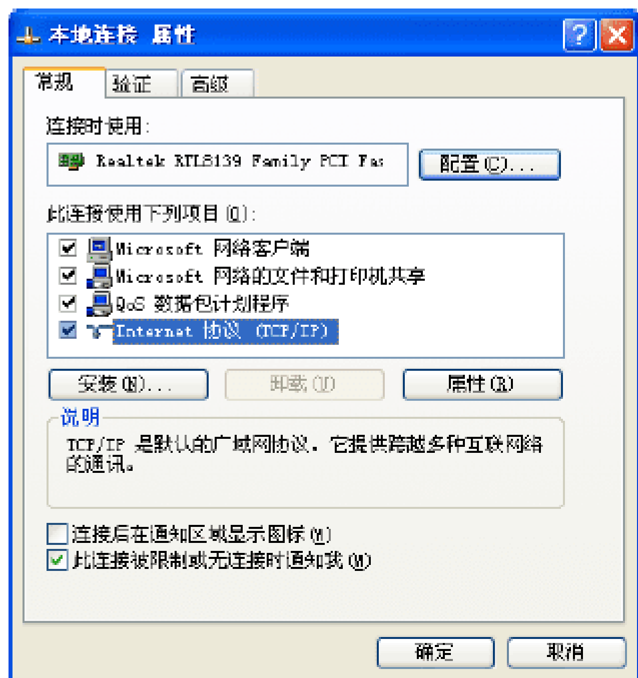


图 1-1 “本地连接属性”对话框

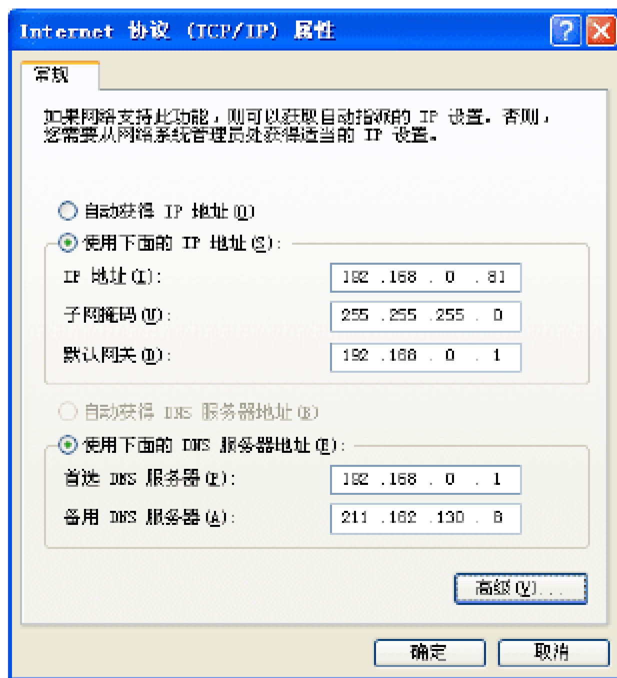


图 1-2 “Internet 协议(TCP/IP)属性”对话框

注意：

在为本地计算机设置 IP 地址时，某些特殊的 IP 地址因为在网络中具有特别的作用，因此不能使用。这些 IP 地址包括 127.X.X.X、0.X.X.X、224.X.X.X~255.X.X.X。

1.2.2 安装与配置 IIS

ASP 程序是运行于网络服务器端的一种应用程序，想要正常运行 ASP 程序，还需要在完成 Internet 服务器 IP 地址的设置工作后建立 ASP 的运行环境。常用的支持 ASP 的网络服务器有 PWS(Personal Web Server)和 IIS(Internet Information Server)。因为应用 PWS 的 Windows 95/98 操作系统目前已被淘汰，下面将重点介绍在 Windows 2000/XP 操作系统中安装与配置 IIS 的方法。

1. 安装 IIS

IIS 内建了 ASP 的编译引擎，如果用户使用 Windows 2000 或更新的 Windows NT 版本操作系统，由于 IIS 是随系统内置的服务组件，因此不需要再重新安装 IIS。但如果用户计算机上安装的是 Windows 2000/XP Professional(专业版)，则仍需要手动安装并配置 IIS 服务。

在 Windows 2000/XP 操作系统上，通常选择 IIS 5.0/5.1 作为 ASP 得以运行的网络服务器。在安装时，用户可以使用 Windows 安装光盘进行设置。

【练习 1-2】在 Windows 2000/XP 系统中安装 IIS 服务。

(1) 选择“开始”|“设置”|“控制面板”命令，在打开的“控制面板”窗口中双击“添加/删除程序”图标，打开“添加/删除程序”窗口，如图 1-3 所示。

(2) 单击“添加/删除 Windows 组件”按钮，打开“Windows 组件向导”对话框。

(3) 在“Windows 组件向导”对话框的“组件”列表框中选择“Internet 信息服务(IIS)”选项(如图 1-4 所示)，然后单击“下一步”按钮，并在光盘驱动器中放入 Windows 2000/XP 安装光盘，即可开始安装文件和配置系统参数。



图 1-3　“添加/删除程序”窗口

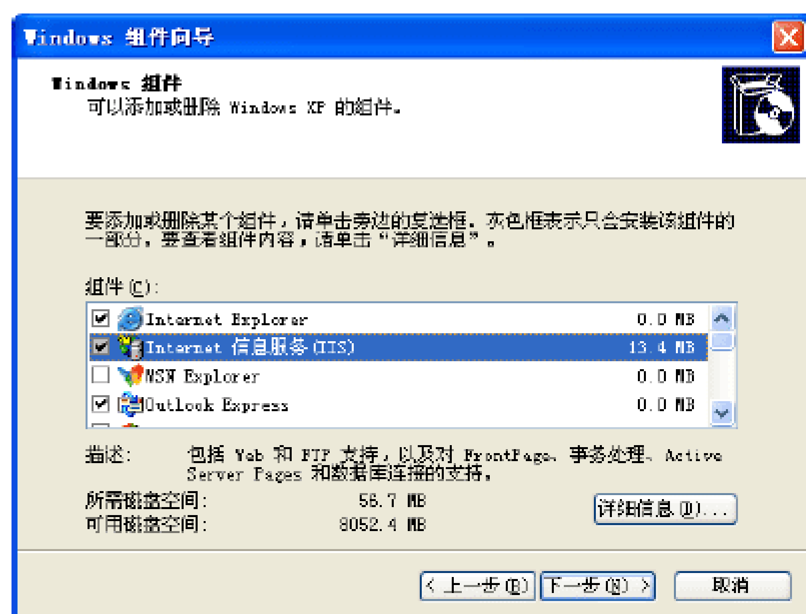


图 1-4　“Windows 组件”对话框

(4) 完成 IIS 组件的安装后，重新启动系统。

2. 配置 IIS

通过在“控制面板”窗口中双击“管理工具”图标，然后在打开的窗口中再次双击“Internet 信息服务”图标(如图 1-5 所示)，可以启动 IIS 的配置主界面(“Internet 信息服务”窗口)。在该界面中右击“默认网站”选项，在弹出的快捷菜单中可以选择“暂停”、“停止”或“启动”命令，来控制默认的 Web 站点的运行状态，也可以选择“新建”|“虚拟目录”命令，发布一个新的 Web 站点，还可以对某个 Web 站点进行“服务器扩展检查”来提高 IIS 的安全性，如图 1-6 所示。



图 1-5　“Internet 信息服务”图标

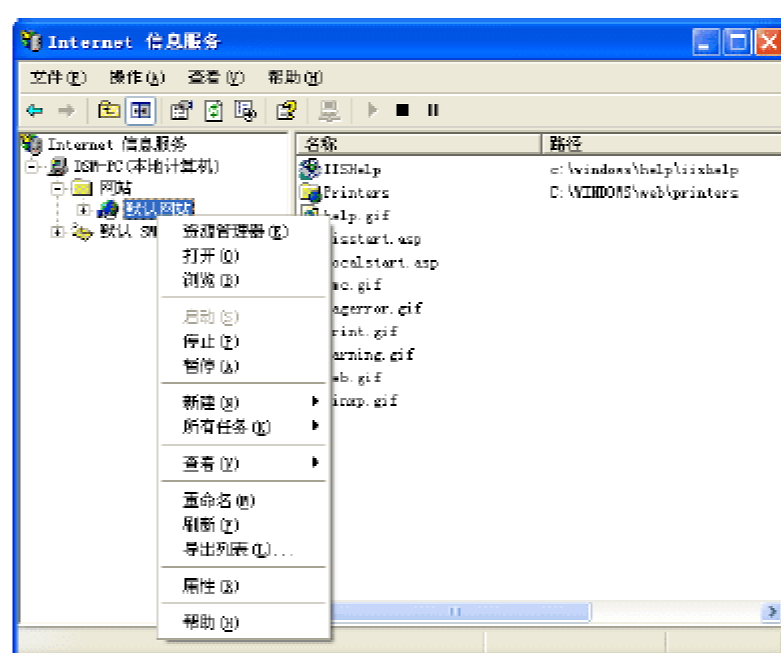


图 1-6　IIS 的配置管理主界面

注意：

IIS 服务提供各种 Internet 服务，例如 FTP 文件传输、发送电子邮件的 SMTP 和网页浏览的网站服务等。当用户使用 IIS 支持 ASP 网站开发时，所有的网页都必须放在 IIS 配置界面的“网站”服务功能下的目录中。通过浏览器浏览特定网页时，IIS 会根据其指定的网址取出对应的文件，并在解析后由 Internet 传送至用户计算机的浏览器中。

如果要进一步配置当前 Web 站点，可以在图 1-6 所示的右键菜单中选择“属性”命令，然后参照下面的方法进行操作。

【练习 1-3】在“Internet 信息服务”窗口中配置 IIS。

(1) 在图 1-6 所示的“Internet 信息服务”窗口中右击“默认网站”选项，在弹出的快捷菜单中选择“属性”命令，打开“默认网站属性”对话框。

(2) 选择“网站”选项卡，可以设置该站点的描述、服务器的 IP 地址和 Web 服务所使用的 TCP 端口等参数，还可以设置连接超时和日志记录等项目，如图 1-7 所示。

(3) 选择“主目录”选项卡，可以设置 Web 站点在服务器上的物理路径，并且可以进行访问权限的设置，如“读取”、“写入”、“目录浏览”、“记录访问”、“脚本资源访问”和“索引资源”属性，如图 1-8 所示。

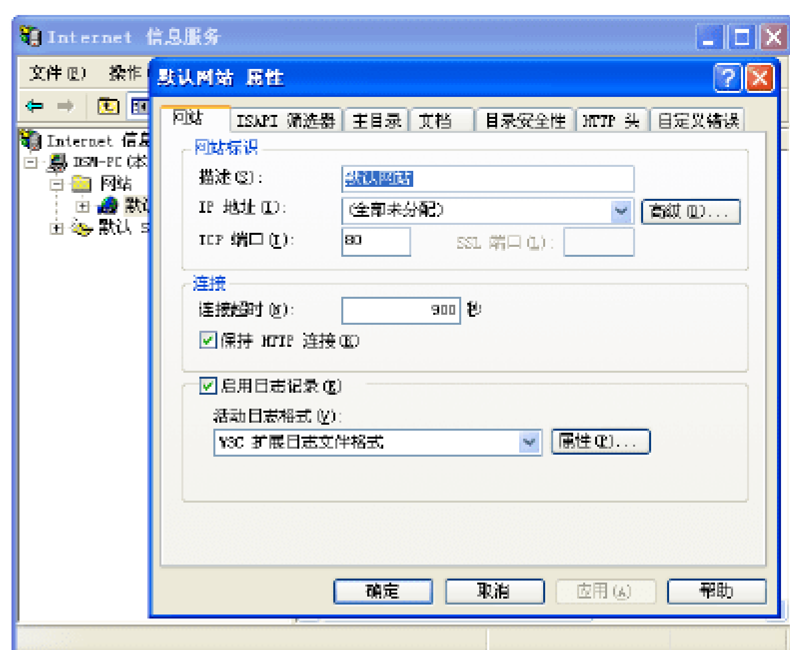


图 1-7　“网站”选项卡

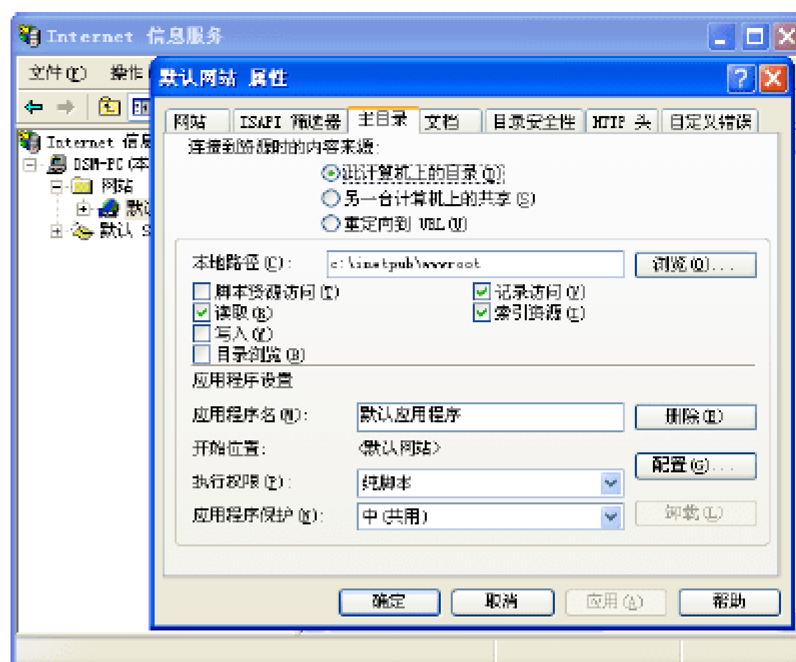


图 1-8　“主目录”选项卡

(4) 在“主目录”选项卡中单击“配置”按钮，在打开的“应用程序配置”对话框中选

择“调试”选项卡，然后选中“启用 ASP 服务器脚本调试”和“启用 ASP 客户端脚本调试”复选框，可以在对 ASP 应用程序进行调试的过程中让系统提供调式帮助，如图 1-9 所示。

(5) 在“默认网站属性”对话框中选择“文档”选项卡(如图 1-10 所示)，可以设置当客户端对该 Web 站点请求连接时默认启动的 HTML 页面或 ASP 应用程序。选择“目录安全性”选项卡可以设置“匿名访问和身份验证控制”和“安全证书”，以确保管理信息系统运行的安全性。

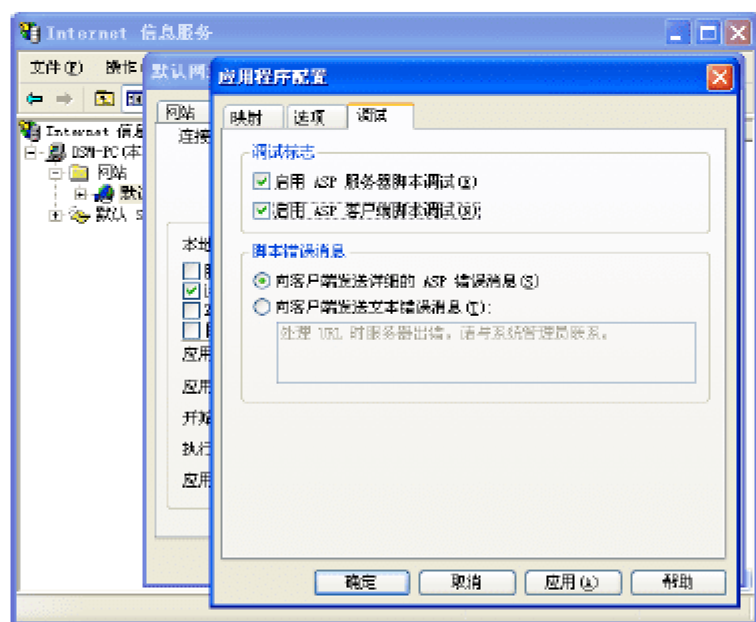


图 1-9　“应用程序配置”对话框

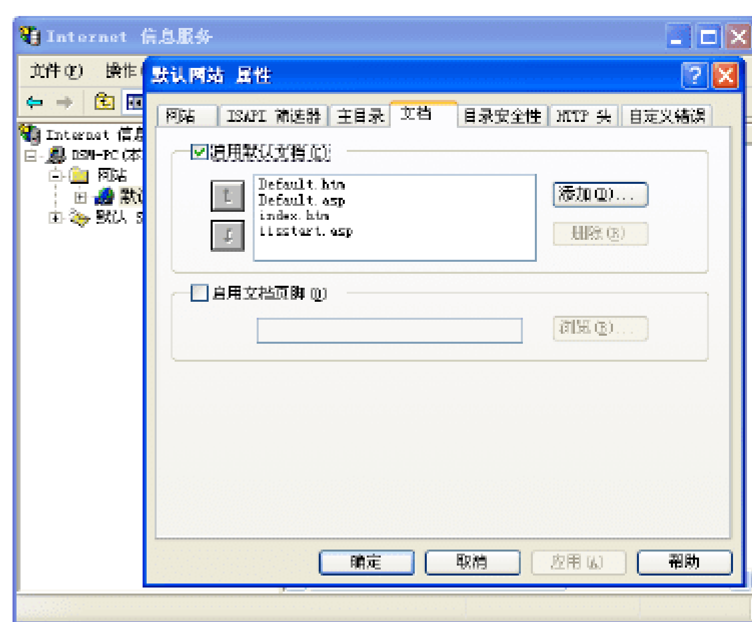


图 1-10　“主目录”选项卡

(6) 完成设置后，在“默认网站属性”对话框中单击“确定”按钮。

3. 测试 IIS

在完成 IIS 的安装与设置后，在浏览器中输入 http://localhost 时，IIS 将自动获取本地计算机上“默认网站”目录下存放的网页文件(Default.asp)，然后将其解析后传送至浏览器显示。用户可以利用 IIS 的这一特点来测试 IIS 服务器工作是否正常。

IIS 默认网站的文件目录列表和与其对应的目录 C:\inetpub\wwwroot 中的内容大致是相互对应的关系。也就是说，在 C:\inetpub\wwwroot 目录中创建的任何.asp 文件和包含.asp 文件的文件夹都可以在“Internet 信息服务”窗口中找到。如果要在浏览器中显示这些文件，只需在地址栏中输入 http://localhost 字符串，并在其后加上其相对路径和文件名即可。例如，要浏览 IIS 安装时在 C:\inetpub\wwwroot 目录中自动产生的网页文件 iisstart.asp，只需要在浏览器的地址栏中输入 http://localhost//iisstart.asp 即可，如图 1-11 所示。

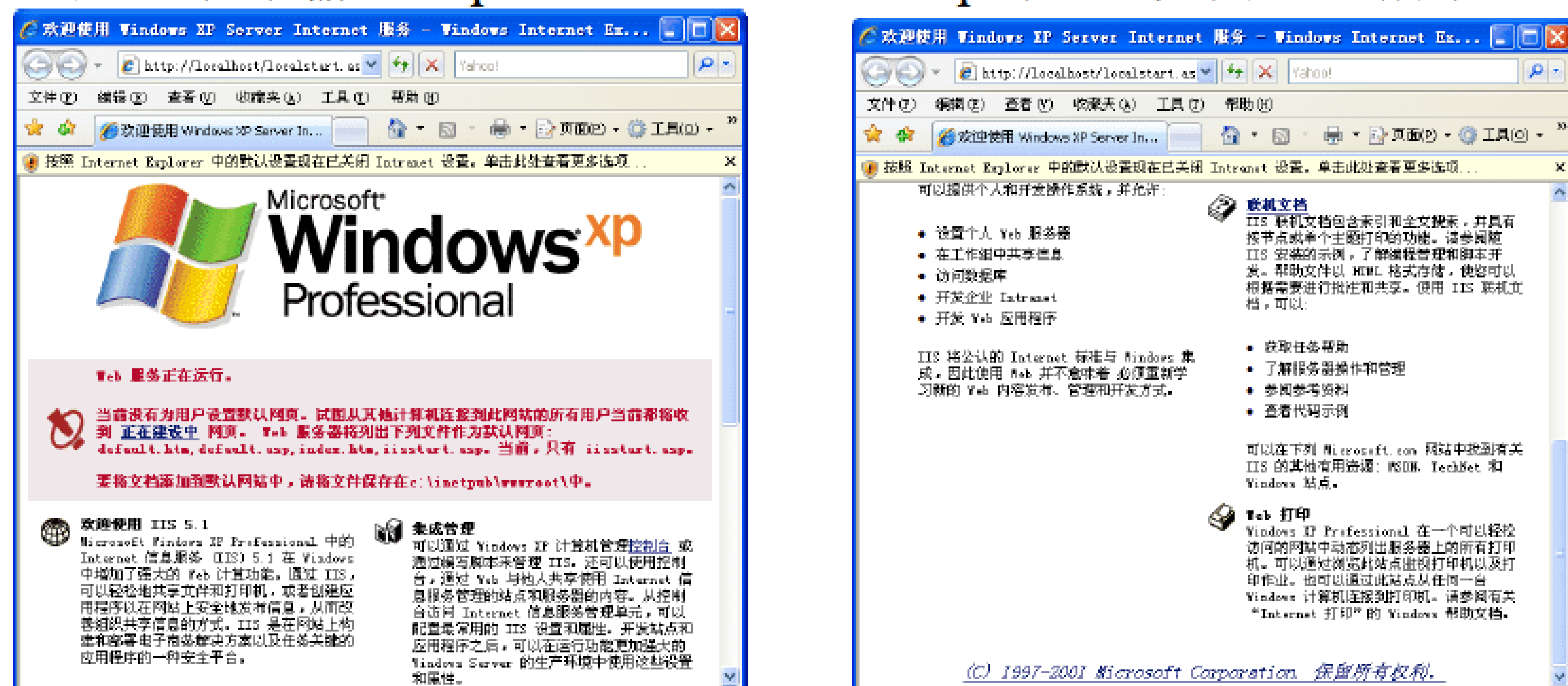


图 1-11　在浏览器中显示 IIS 默认网站文件夹中的 ASP 网页

1.3　创建 ASP 应用程序

ASP 实际上是将标准的 HTML 文件扩展了一些附加特征，它像标准的 HTML 文件一样包含 HTML 语句并且最终在浏览器上解释并显示。ASP 文件的后缀为.asp，其中包含实现动态功能的 VBScript 或 JScript 语句，如果去掉那些 VBScript 或 JScript 语句，它和标准的 HTML 文件没有任何区别。

1.3.1　创建虚拟目录

在创建 ASP 应用程序之前，若要从主目录外的目录发布网页，则可通过创建虚拟目录来进行。虚拟目录是指物理上未包含在主目录中的目录，但浏览器却认为该目录包含在主目录中。

【练习 1-4】在 C:\Inetpub\wwwroot 目录中创建为一个名为 test 的虚拟目录。

(1) 在如图 1-6 所示的“Internet 信息服务”窗口中，右击左侧的“默认网站”选项，在弹出的快捷菜单中选择“新建”|“虚拟目录”命令，打开“虚拟目录创建向导”对话框，如图 1-12 所示。

(2) 单击“下一步”按钮，输入虚拟目录别名(例如 test)，也就是在访问网页时需要输入的名称，如图 1-13 所示。别名一般比目录的路径名短，更便于用户输入，也更安全。

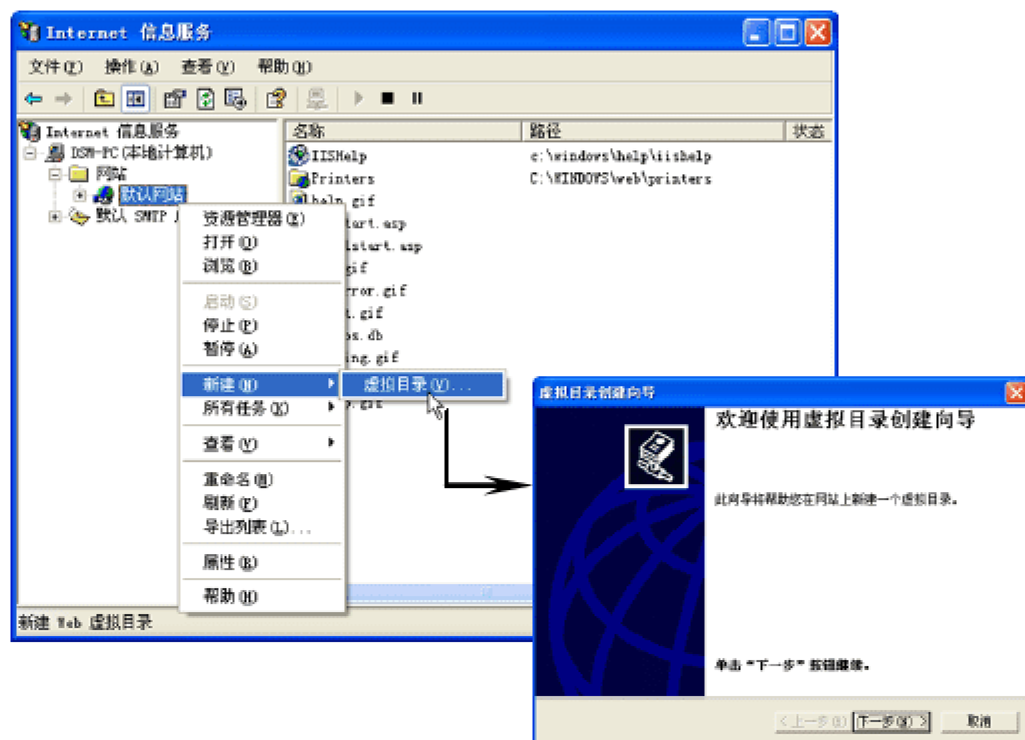


图 1-12　打开“虚拟目录创建向导”对话框

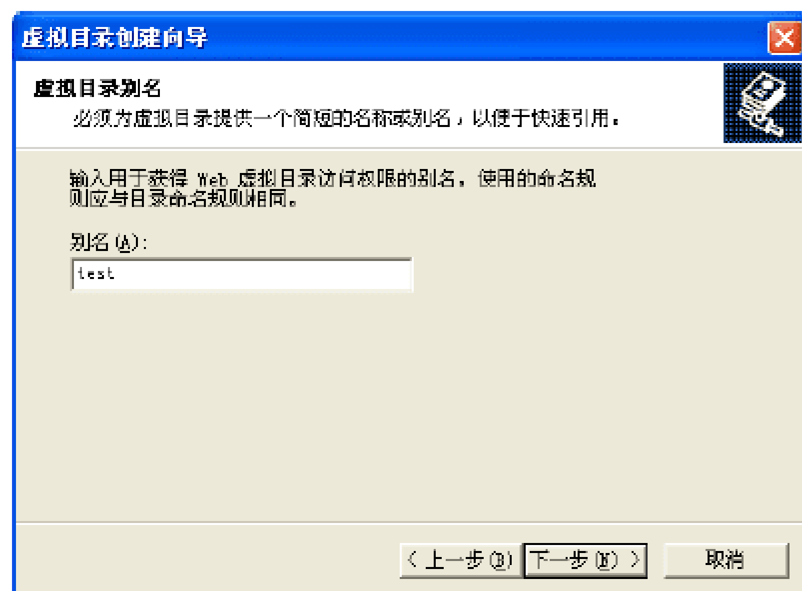


图 1-13　输入虚拟目录名称

(3) 单击“下一步”按钮，输入虚拟目录的路径 C:\Inetpub\wwwroot，如图 1-14 所示，以后就可通过虚拟名称来访问该目录中的文件了。

(4) 单击“下一步”按钮，进入权限设置步骤，为了保证网站的安全，只需选择前 3 个复选框(“读取”、“运行脚本”和“执行”复选框)就可以了，如图 1-15 所示。

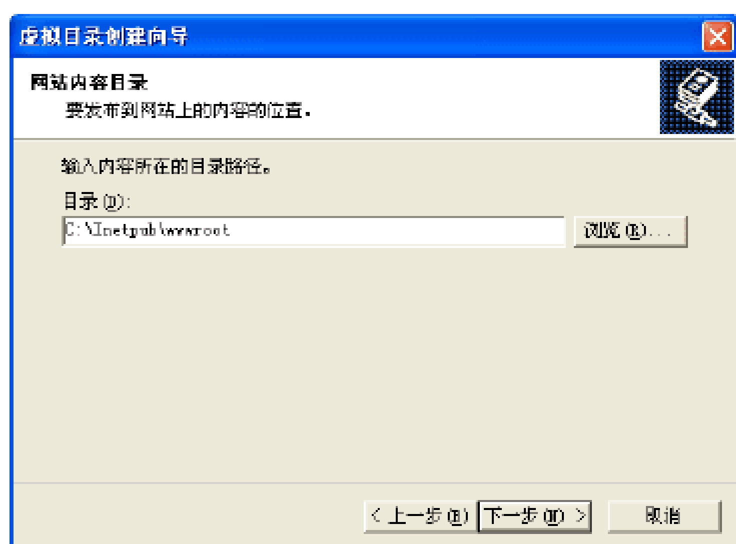


图 1-14　“虚拟目录创建向导”对话框

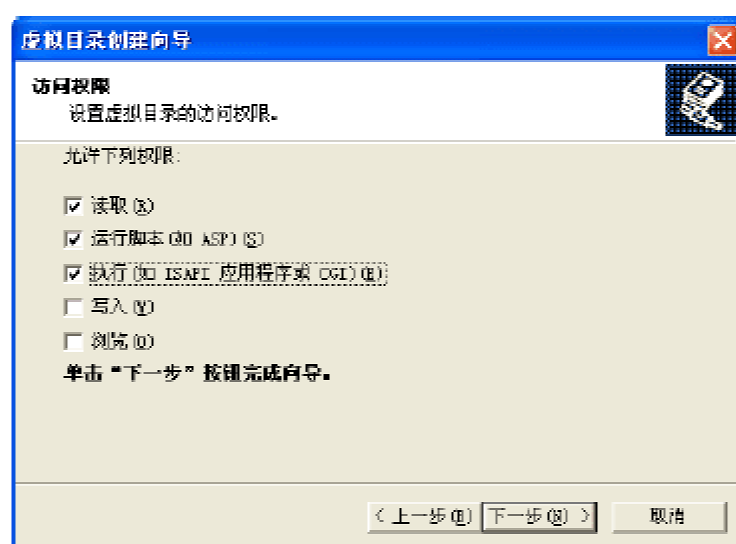


图 1-15　访问权限设置

(5) 单击“下一步”按钮，在打开的“已成功完成虚拟目录创建向导”对话框中单击“完成”按钮，完成虚拟目录创建。

1.3.2　创建 ASP 网页

下面以【练习 1-4】建立的虚拟目录为基础，通过实例介绍创建 ASP 网页的方法和编写 ASP 程序的注意事项。

1. 创建 ASP 应用程序

在 ASP 程序中，脚本通过分隔符将文本和 HTML 标记区分开来。ASP 用分隔符<%和%>来包括脚本命令。ASP 文件中一般包含 HTML 标记、VBScript 或 JScript 语言的程序代码以及 ASP 语法。

【练习 1-5】以【练习 1-4】创建的虚拟目录 test 为基础，使用记事本编写一个查看系统时间的 ASP 程序。

(1) 选择“开始”|“所有程序”|“附件”|“记事本”命令，打开记事本。

(2) 输入以下代码(如图 1-16 所示)：

```
<Html>
<Body>
您访问本页面的时间是<%=Time()%>！
</Body>
</Html>
```

(3) 选择“文件”|“另存为”命令，将该文件命名为 test_time.asp，并保存至 C:\Inetpub\wwwroot 文件夹中。

注意：

本书中所有 ASP 应用程序都是通过步骤(1)～(3)来创建的，并保存至 C:\Inetpub\wwwroot 文件夹中，因此，在后面的章节中只给出相应 ASP 程序的详细代码。

(4) 启动 IE 浏览器，在地址栏中输入 http://localhost/test/test_time.asp，按下 Enter 键，其运行效果如图 1-17 所示。

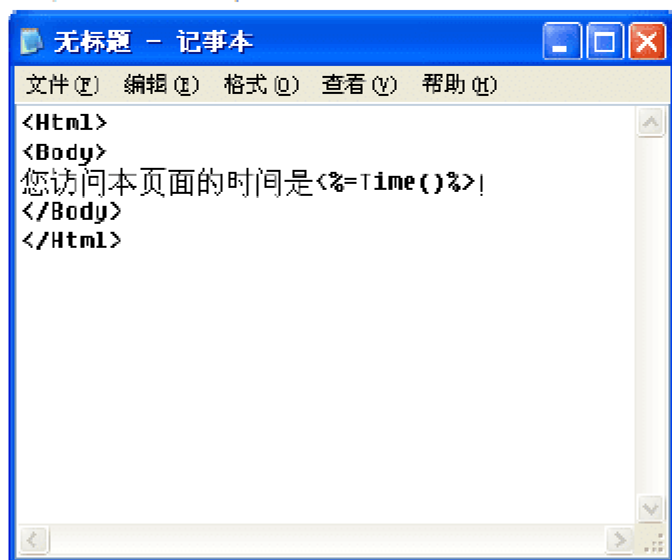


图 1-16　输入代码

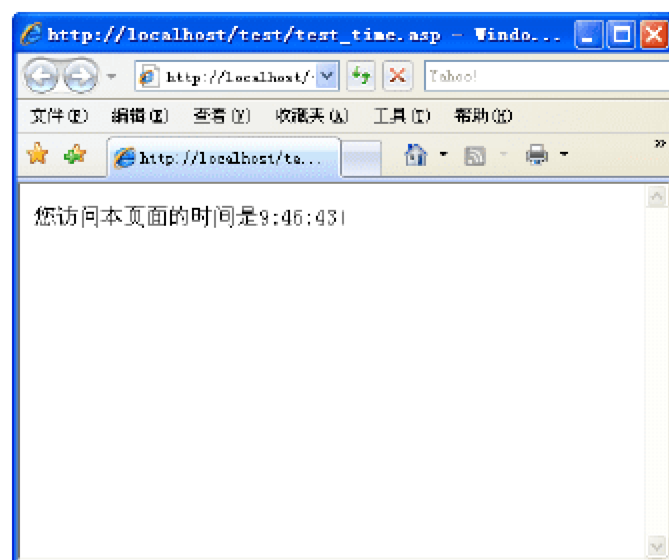


图 1-17　查看系统时间的网页效果

注意：

【练习 1-5】所示的例子是在标准 HTML 页面代码中嵌入了 VBScript 代码，<%和%>符号之间的内容即是 VBScript 代码，Time()运行的结果就是显示当前时间。执行时，Web 服务器将<%=Time()%>替换为当前时间，然后将结果返回到浏览器中。

(5) 右击 C:\Inetpub\wwwroot 文件夹中的 test_time.asp 文件，在弹出的快捷菜单中选择“打开方式”|“记事本”命令，打开记事本，如图 1-18 所示。

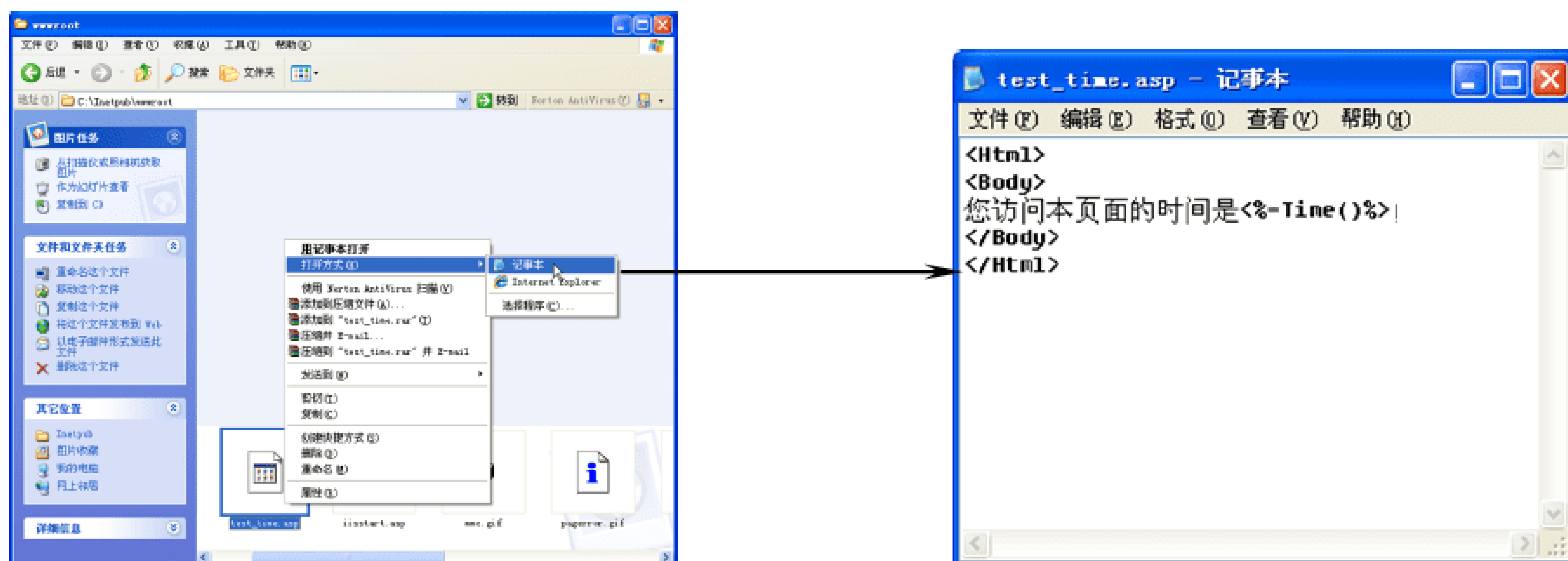


图 1-18　打开 ASP 文件

(6) 在记事本中，修改代码如下所示(如图 1-19 所示)：

```
<Html>
<Body>
<%For I=1 To 6%>
<Font Size="<%=I%>">您访问本页面的时间是<%=Time()%>！</Font><Br>
<%Next%>
</Body>
</Html>
```

(7) 选择“文件”|“另存为”命令，命名文件为 test_time1.asp，并保存至 C:\Inetpub\wwwroot 文件夹中。

(8) 在浏览器的地址栏中输入 http://localhost/test/test_time1.asp，按下 Enter 键，其运行效果如图 1-20 所示。

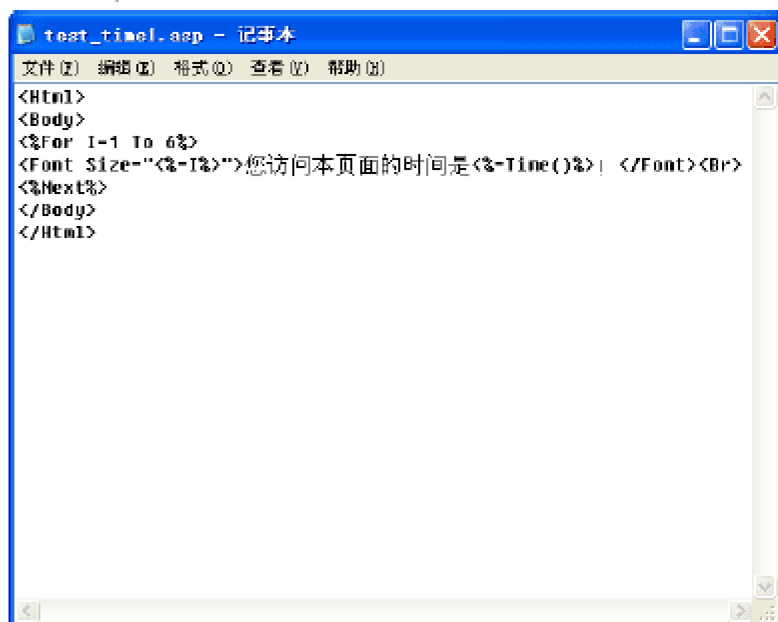


图 1-19　修改代码

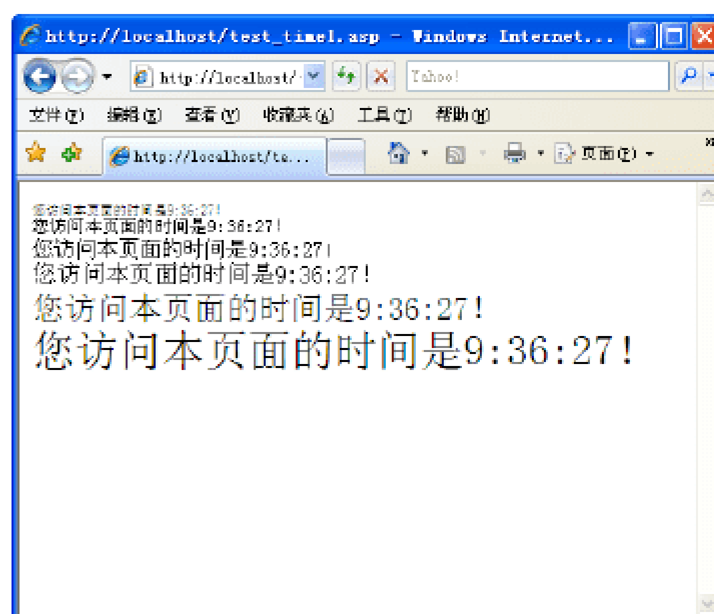


图 1-20　修改后的效果

2. 编写 ASP 程序的注意事项

在编写 ASP 程序时，需要注意以下事项。

(1) 在 ASP 程序中，字母不分大小写。用户可根据自己的习惯，自由选择代码的输入形式。

(2) 在 ASP 中，<%和%>符号的位置是相对随便的，可以和 ASP 语句放在一行，也可以单独成为一行。例如下面 3 种写法效果都是一样的：

```
<%For I=1 To 6%>
```

```
<%
For I=1 To 6
%>
```

```
<%For I=1 To 6
%>
```

(3) ASP 语句必须分行写，不能将多条 ASP 语句写在一行里，也不能将一条 ASP 语句写在多行里。例如，下面的两个例子都是错误的：

```
<% a=2 b=3 %>
```

```
<%
a=
2
%>
```

(4) 如果一条 ASP 语句过长，需要换行时可采用两种方法。一种方法是可以不用 Enter 键分开，而是直接书写，使之自动换行；另一种方法是用 Enter 键将该语句分成多行，只是必须在每行末尾(最后一行除外)加一个下划线，如下面的例子：

```
<%if time <#12:00# and time>=#00:00:00# then
 strGreeting="欢迎来访！这里是我们最新制作的网站_
早上好！欢迎你在参观后提出宝贵的意见" %>
```

(5) 在 ASP 中，使用 REM 或“'”符号来标记注释语句，运行时 ASP 不执行注释语句。在代码中添加注释主要是为了方便自己和别人阅读程序代码，如下面的例子：

```
<%
REM 这是一条注释语句！
'这是另一条注释语句！
%>
```

另外，在编辑 ASP 程序代码时，要养成良好的书写习惯，例如可以为代码添加上恰当的缩进。这样，以后自己和别人阅读起来都方便一些，否则代码很不容易读懂，缩进的方法可以参考本书中的代码书写样式。

1.4　习　　题

1.4.1　填空题

(1) ASP 主要为 HTML 编写人员提供了在服务器端运行脚本的环境，使 HTML 编写人员可以利用___________和___________或其他第三方脚本语言来创建 ASP，实现有动态内容的网页。

(2) ASP 程序的脚本不是在客户端运行的，传送到浏览器上的 Web 页是在___________上生成的。

(3) IIS 允许在一台计算机上创建多个 Web 站点，这些站点可以共同使用一个 IP 地址同时提供信息发布服务。它的实现方法是为不同网站指定一个不同的___________来加以区分。

(4) ASP 文件的后缀为___________。

(5) ASP 用分隔符___________来包括脚本命令。

1.4.2　选择题

(1) 下面关于动态网页的说法不正确的是(　　)。

A. 可从 HTML 的表单中接收信息并且存到数据库中

B. 可根据不同访问者显示不同内容，创建个性化主页

C. 可跟踪用户网站上的活动信息并且存入日志文件

D. 需要浏览器执行网页中动态效果的程序

(2) 在 ASP 文件中，不可以包含以下(　　)内容。

A. HTML 标记　　B. VBScript 或 JScript 语言的程序代码

C. ASP 语法　　D. 声音、图像等多媒体

(3) 下面关于编写 ASP 程序，说法不正确的是(　　)。

A. <%和%>符号必须和 ASP 语句放在一行

B. ASP 语句必须分行写，不能将多条 ASP 语句写在一行里

C. 使用 REM 或“'”符号来标记注释语句

D. 在 ASP 程序中，字母不分大小写

1.4.3　问答题

(1) 简述 ASP 的特点及好处。

(2) 简述当客户请求访问一个 ASP 网页时，服务器相应的处理工作流程。

(3) 试说明如何在 Windows XP 下安装 IIS 服务。

(4) 试说明如何设定 IIS 的默认浏览文件。

1.4.4　操作题

(1) 参照本章【练习 1-1】～【练习 1-4】所示的操作，在当前计算机上配置一个 ASP 开发环境，并创建一个名为 Test 的虚拟目录。

(2) 参考本章【练习 1-5】的操作，在记事本中输入下面的程序代码，然后在上题创建的 ASP 环境中运行。完成操作后，思考代码各语句的含义和用法，并通过调整当前系统的时间查看其运行效果。

```
<html>
<body>
当前时间是<%=Time()%><br>
<%
if time <#12:00# and time>=#00:00:00# then
    response.write"早上！"
ElseIf Time<#19:00:00# and time >=#12:00:00# then
    response.write"下午！"
Else
    response.write"晚上！"
End if
%>
</body>
</html>
```

第2章　网页设计基础

掌握 HTML 语言是制作网页的基础。HTML 文件包含了所有显示在网页中的内容信息，例如文本的位置，网页的显示模式以及网页图片、动画和声音等资源的设置。本章将通过介绍 HTML 语言的各种标记和语法，介绍制作网页的基础知识。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 了解使用 HTML 语言设计静态网页的方法
- 掌握编辑网页文本格式的方法
- 掌握创建超链接的方法
- 掌握表格和列表的使用方法

2.1　认 识 网 页

网页是用一种叫做 HTML 的标记语言来表示的，它将要表示的信息语言用 HTML 表述出来，这就形成了 HTML 代码。而浏览器的主要工作就是将这些标记语言“翻译”过来，并按照定义的格式显示，这就是最终看到的网页。

2.1.1　网页的基础知识

Internet 是从 Interconnected Networks 延伸而来的，是跨国界的网络。Internet 把世界各地数以千万计的计算机和传输线路连接在一起构成一个网络。通过它可以交换信息、共享资源，并以此为基础实现各种计算机通信应用项目。在 Internet 中，网页是它的重要组成部分，本节首先介绍一些与网页相关的名词和概念。

1. 万维网(WWW)

WWW(World Wide Web)即环球信息网，也可以称为 Web，中文名字为“万维网”。用户在使用浏览器访问 Web 的过程中，无需关心一些技术性的细节即可得到丰富的信息资料。WWW 是 Internet 上发展最快，目前使用最广泛的一种服务。

简单地说，WWW 是漫游 Internet 网络的工具，它把 Internet 上不同地点的相关信息聚集起来，通过 WWW 浏览器(例如 IE，Internet Explorer)检索，无论用户所需的信息在什么地方，只要浏览器为用户检索到之后，就可以将这些信息(文字、图片、动画和声音等)“提取”到用户的计算机屏幕上。

2. 超文本传输协议(HTTP)

HTTP(HyperText Transfer Protocol)即超文本传输协议，是 WWW 服务器上使用的最主要协议。通过这一跨平台的通信协议，在 WWW 任何平台上的计算机都可以阅读远方服务器(Server)上的同一文件。

HTTP 协议经常用来在网络上传送 Web 页。当用户以 http://开始一个超链接的名字时，就是告诉浏览器去访问使用 HTTP 协议的 Web 页。

HTTP 协议不仅能保证正确传输超文本文档，还可以确定传输文档中的哪一部分，以及哪部分内容首先显示(如文本先于图形)等。

3. 统一资源定位器(URL)

URL(Uniform Resource Locator)即统一资源定位器，它使用数字和字母来代表网页文件在网上的地址。URL 好比 Internet 的门牌号码，可以帮助用户在 Internet 的信息海洋中查找到所需要的资料。

Web 上所能访问的资源都有唯一的 URL。URL 包括所用的传输协议、服务器名称、文件的完整路径。例如，在浏览器的 URL 处输入 http://www.sohu.com/index.html 就可以访问搜狐网站的主页。

URL 的第一部分 http://表示要访问的资源类型。其他常见资源类型中，ftp://表示 FTP 服务器，gopher://表示 Gopher 服务器，new://表示 Newgroup 新闻组。

第二部分 www.sohu.com 是主机名，它说明了要访问服务器的 Internet 名称。其中，www 表示要访问的文件存放在名为 www 服务器里，多数公司都有指定的服务器作为对外的网上站点，叫做 www；sohu 则表示了该网站的名称；.com 则指出了该网站的服务类型。

目前，常用的网站服务类型的含义如下：.com 特指事务和商务组织；.edu 表示教育机构；.gov 表示政府机关；.mil 表示军用服务；.net 表示网关，由网络主机或 Internet 服务提供商决定；.org 一般表示公共服务或非正式组织。

注意：

另外，有些域名后面会带有本国和地区的域名。例如，新浪网址 http://www.sina.com.cn 中的 cn 代表该网站属于中国。另外，au 代表澳大利亚，ca 代表加拿大，fr 代表法兰西，uk 代表英国，jp 代表日本等。

第三部分/index.html 表示要访问主机的哪一个页面文件，可以把它理解为该文件存放在服务器上的具体位置。

4. 超文本标记语言(HTML)

HTML(HyperText Mark-up Language)即超文本标记语言，是 WWW 上通用的描述语言。HTML 语言主要是为了把存放在一台计算机中的文本或图形与另一台计算机中的文本或图形方便地联系在一起，形成有机的整体。

5. HTML 标记

HTML 标记用于修饰、设置 HTML 文件的内容及格式。用户只需输入文件内容和必要的标记，文件内容在浏览器窗口内就会按照标记定义的格式显示出来。一般情况下，HTML 标记使用下列格式：

```
<标记>文件内容</标记>
```

标记需要填写在一对尖括号“< >”内，它们通常是英文单词的首字母或缩写。一般情况下标记是成对出现的。结束标记是在标记的前面添加斜杠“/”。

在书写标记时，英文字母的大小写或混合使用大小写都是允许的，如 HTML、html 和 Html 的作用和效果都是一样的。

标记内可以包含一些属性。标记属性可由用户设置，否则将采用默认的设置值。属性名称出现在标记的后面，并且以空格进行分隔。如果标记具有多个属性，那么不同的属性名称之间将以空格隔开。其格式如下：

```
< 标记名字  属性 1  属性 2  属性 3  …  >
```

HTML 对属性名称的排列顺序没有特别的要求，用户可根据个人的爱好，在标记之后排列所需的属性名称。另外，标记的属性值须使用双引号或单引号括起来。

注意：

HTML 文件可以使用一个简单的文本编辑器创建，例如 Windows 操作系统中自带的记事本程序。但是文件编辑完成进行保存时，必须以 htm 或 html 为扩展名。以后，就可以使用 Web 浏览器(如 IE 浏览器)来直接打开 HTML 文件进行浏览和查看了。

2.1.2　网页的基本结构

HTML 文件通常由 3 部分组成，即起始标记、网页标题和文件主体。其中，文件主体是 HTML 文件的主要部分与核心内容，它包括文件所有的实际内容与绝大多数的标记符号。

在 HTML 文本中，有一些固定的标记要放在每一个 HTML 文件里。HTML 文件的总体结构如下所示：

```
<Html>
<Head>
网页的标题及属性
</Head>
<Body>
文件主体
</Body>
</Html>
```

1. 添加起始标记

<Html>标记用于 HTML 文档的最前面，用来标识 HTML 文档的开始。而</Html>标记恰恰相反，它放在 HTML 文档的最后面，用来标识 HTML 文档的结束，两个标记必须一起使用。通过对这一对特殊标记符号的读取，浏览器才可以判断目前正在打开的是网页文件，而不是其他类型的文件。

<Html>标记的起始和结束标记符号都是可选的，但用户应该养成在文件中使用<Html>标记的习惯，每次编写 HTML 文件之前都应该首先在网页内添加<Html>…</Html>标记对，然后再在标记对之间加入网页的内容。

2. 设置文件头

<Head></Head>标记对构成 HTML 文档的开头部分，在此标记对之间可以使用<Title></Title>、<Script></Script>等标记对。这些标记对都是描述 HTML 文档相关信息的标记对，<Head></Head>标记对之间的内容不会在浏览器的窗口内显示出来，两个标记必须一起使用。常用的文件头标记如表 2-1 所示。

表 2-1　常用的文件头标记

标　记	说　明
Title 标记	<Title>和</Title>标记标明 HTML 文件的标题，是对文件内容的概括。一个好的标题能使用户从中判断出该文件的大概内容。文件的标题一般不会显示在文本窗口中，而以窗口的名称显示在标题栏中。<Title></Title>标记对只能放在<Head></Head>标记对之间。例如： <Title>我的网页</Title>
Isindex 标记	<Isindex>标记主要用来定义查询的表单，表明允许用户通过提供关键字的方式来查找索引内容。在阅读文件时，不管用户采用什么样的方式，当前文件所描述的数据均可采用索引查询方式进行搜索
Base 标记	<Base>标记用于设定超链接的基准路径。使用这个标记，可以大大简化网页内超链接的编写。用户不必为每个超链接输入完整的路径，而只需指定它相对于<Base>标记所指定的基准地址的相对路径即可。该标记包含参数 Href，用于指明基准路径，其用法如下： <Base href="URL">
Link 标记	<Link>标记表示超链接，在 HTML 文件的<Head>标记中可以出现任意数目的 Link 标记。它也包含参数 Href。<Link>标记可以定义含有链接标记的文件与 URL 中定义文件之间的关系。 <Link>标记通常用来显示作者身份、相关检索及术语、旧的或更新的版本、文件等级、相关资源等。Rel 参数用来定义 HTML 文件及 URL 之间的关系，Rev 参数用来定义 URL 和 HTML 文件之间的关系。该标记用法如下： <link rev="RELATIONSHIP"　rel="RELATIONSHIP" href="URL">

(续表)

标　　记	说　　明
Meta 标记	<Meta>标记用来指明与文件内容相关的信息。每一个该标记指明一个名称或数值对。如果多个<Meta>标记使用了相同的名称，其内容便会合并成一个用逗号隔开的列表，也就是和该名称相关的值。<Meta>标记的主要属性包括如下内容。 ● Http-Equlv：把标记放到 HTTP 头域之中。HTTP 服务器可使用该信息处理文件，特别是它可在对这个文件请求的响应中包含一个头域。标题名取自 Http-Equiv 属性值，而标题值则取自 Content 属性值。 ● Name：指明名称或数值对的名称。如果没有，则由 Http-Equiv 给出名称。 ● Content：指明名称或数值对的值，一般为 text/html。 ● Charset：指明网页所使用的基本字符集，一般为 GB2312，即标准简体中文。 Meta 标记的一般用法如下： <Meta http-equiv="Content-Type" content="text/html;charset=gb2312">

2.2 创 建 网 页

本节将通过简单的操作实例，详细介绍在 Windows 系统附带的记事本中，通过输入 HTML 代码创建网页的方法。

2.2.1 编辑网页正文

正文是网页的核心内容，用户可使用 HTML 语言在网页内对正文进行划分段落、插入标题、修改字体、设置字号等操作。另外，HTML 还允许对正文应用不同的字形和效果。

1. 输入网页正文标题

一般文章都有标题、副标题、章和节等结构，HTML 也提供了相应的标题标记<Hn>，其中 n 为标题的等级。HTML 提供 6 个等级的标题，n 越小，标题字号就越大。

【练习 2-1】使用标题标记改变文本格式，创建如图 2-1 所示的网页。

(1) 创建一个名为 test.html 的网页，代码如下所示(如图 2-2 所示):

```
<H1>标题 1</H1>
<H2>标题 2</H2>
<P>这是一行没有设置标题格式的正文文本</P>
<H5>标题 5</H5>
```

<H6>标题 6</H6>

(2) 运行 test.html 网页，效果如图 2-1 所示。

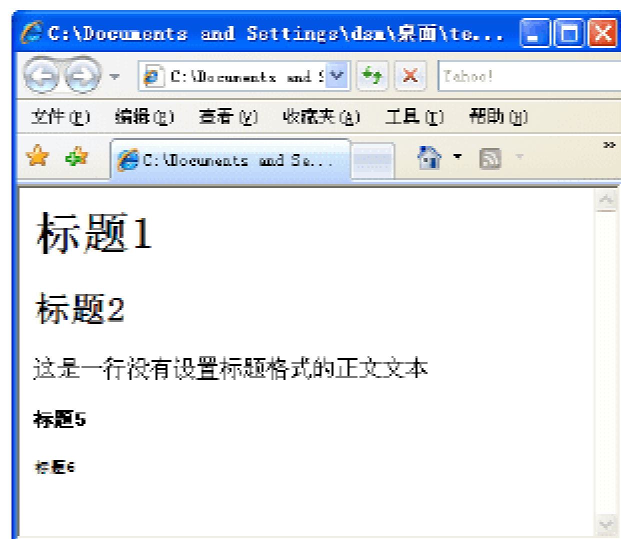


图 2-1　test.html 的运行效果

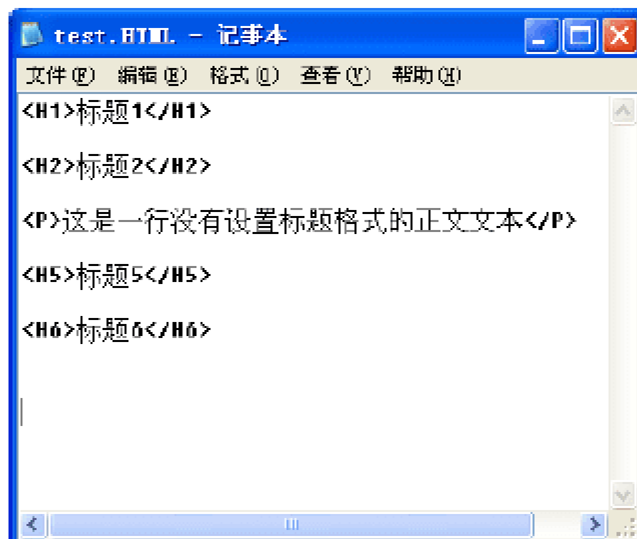


图 2-2　test.html 的代码

注意：

用户除了使用本例所介绍的方法测试网页效果以外，还可以参考本书第 1.3.1 节【练习 1-5】所介绍的操作，将本例中的代码以扩展名 ASP(test.asp)保存至 C:\Inetpub\wwwroot 文件夹中，然后通过在浏览器中输入 http://localhost/test/test.asp 来测试网页效果。

2. 为页面文字划分段落

<P></P>标记对用来创建一个段落，在此标记对之间加入的文本将按照段落的格式显示在浏览器上。HTML 将多个空格以及回车等效为一个空格，HTML 的分段完全依赖于分段标记<P>。

另外，<P>标记还可以使用 Align 属性，用来说明对齐方式，语法是<P Align="对齐方式"></P>。Align 的值可以是 Left(左对齐)、Center(居中)和 Right(右对齐)3 个值中的任何一个。如<P Align="Center"></P>表示标记对中的文本使用居中对齐方式。

【练习 2-2】使用段落标记，并设置不同的属性，创建如图 2-3 所示的网页。

(1) 创建一个名为 test2.html 的网页，其代码如下所示(如图 2-4 所示):

```
<P Align="Left">文字左对齐。</P>
<P Align="Center">文字居中对齐。</P>
<P Align="Right">文字右对齐。</P>
<P >上面的三段文字被 P 标记设置了段落格式。</P>
```

(2) 运行 test2.html，网页效果如图 2-3 所示。

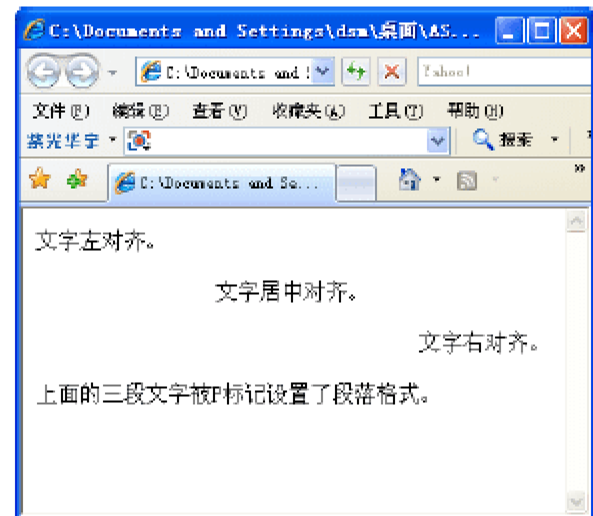


图 2-3　test2.html 的运行效果

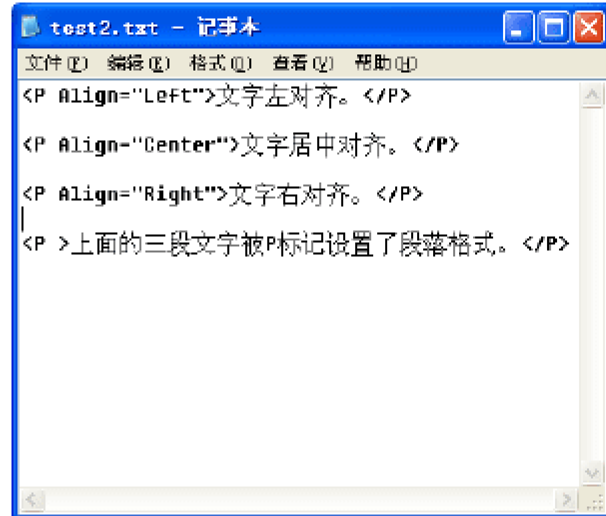


图 2-4　test2.html 的代码

注意：

标记用来创建一个回车换行。如果
标记处在<P></P>标记对的外面，将创建一个大的回车换行，即
标记前面和后面的文本的行与行之间的距离比较大。若处在<P></P>标记对的中间，则
标记前面和后面文本的行与行之间的距离将比较小。

3. 设置页面文字的字体和字号

<Font></Font>标记对用来设置文字字体，它的 Face 属性指定浏览器所显示文本的字体类别，而 Size 和 Color 属性则可以对输出文本的字体大小、颜色进行随意地改变。

用户在使用 Font 标记的 Face 属性设置文本字体时，可指定一个字体列表，如果浏览器不支持第一种字体，就会依次使用第二种、第三种等后续字体显示网页内容。如下例：

```
<Font Face="宋体,仿宋体,隶书">我要显示的汉字</Font>
```

Size 属性用来改变字体的大小；Color 属性则用来改变文本的颜色，颜色的取值是十六进制 RGB 颜色码或 HTML 语言给定的颜色常量名。

4. 设置页面文字的强调及加重等效果

下面的标记对用于设置文本的强调、加重等效果，其用法和前面的标记一样，差别在于输出的文本字体不太一样。

- <Tt></Tt>标记对：用来输出打字机风格字体的文本。
- <Cite></Cite>标记对：用来输出引用方式的字体，通常是斜体。
- <Em></Em>标记对：用来输出需要强调的文本(通常是斜体加粗体)。
- <Strong></Strong>标记对：用来输出加强显示效果的文本(通常也是斜体加粗体)。

【练习 2-3】参考上面所介绍的内容，为网页中的文字设置不同的字体，创建如图 2-5 所示的网页。

(1) 创建一个名称 test3.html 的网页，其代码如下所示(如图 2-6 所示):

```
<Body text="blue">
<H1>最大的标题</H1>
<H3>使用 h3 的标题</H3>
<H6>最大的标题</H6>
<P><B>黑体字文本</B> </P>
<P><I>斜体字文本</I> </P>
<P><U>下加一划线文本</U> </P>
<P><Tt>打字机风格的文本</Tt></P>
<P><Cite>引用方式的文本</Cite></P>
<P><Em>强调的文本</Em></P>
<P><Strong>加重的文本</Strong></P>
<P><font Size="+1" Color="red">Size 取值“+1”、Color 取值“red”时的文本</font></P>
```

(2) 运行 test3.html 网页，效果如图 2-5 所示。

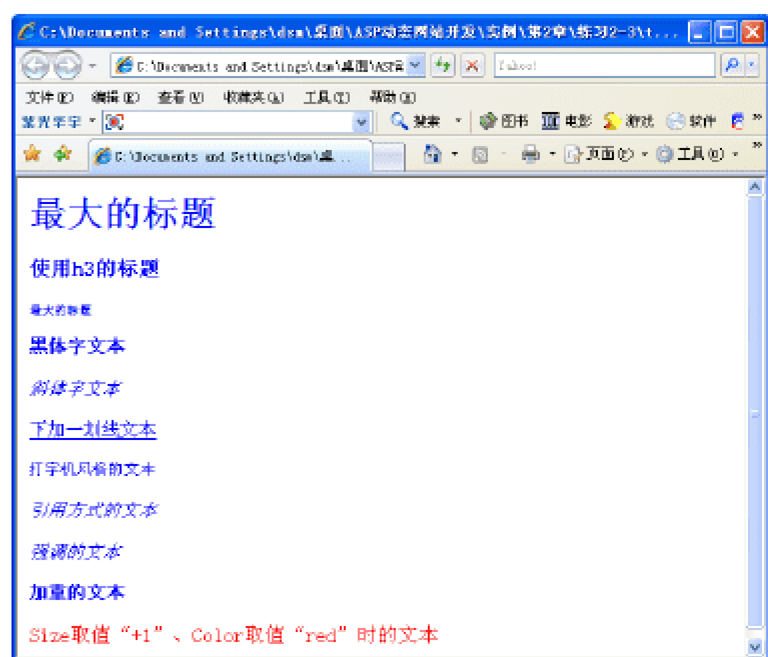


图 2-5　test3.html 的效果

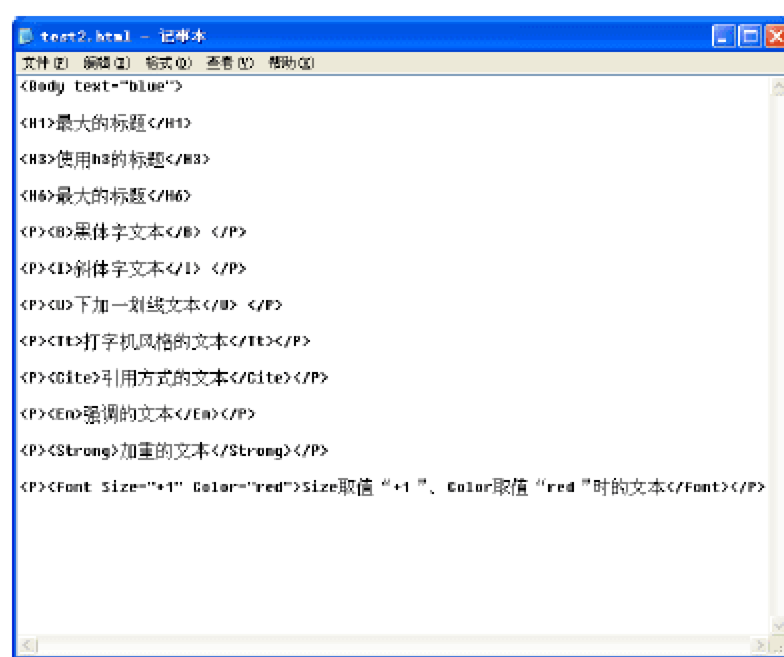


图 2-6　test3.html 的代码

5. 设置页面文字的黑体、斜体和下划线效果

除了正常的字体外，用户还可为文本设置黑体、斜体和下划线等字型。HTML 对这些标记出现的次序没有特别的要求。

- <B></B>标记对：用来使文本以黑体字的形式输出。
- <I></I>标记对：用来使文本以斜体字的形式输出。
- <U></U>标记对：用来使文本以加下划线的形式输出。

2.2.2　插入网页图像

制作网页时，常需要在页面上添加一些图片，因为有时“一图胜千言”。HTML 语言提供了<Img>标记来处理图像的输出。本节将介绍在网页内插入图像文件，设置图像链接的方法，以及通过运用 HTML 标记改变图像的显示尺寸与对齐方式等。

1. 在网页中插入图像

HTML 采用的图像格式有 GIF、JPG 和 PNG 3 种。在网页中插入图像时，须使用 HTML 的<Img>标记，它的使用格式如下：

```
<Img Src="Picname">
```

Src 是 Source(源)英文的缩写，Picname 是希望在网页内显示的图像的 URL。在网页内创建图像链接与文本链接的区别并不大，也需要使用<A>标记，并指明链接目标的 URL，唯一的区别就是在</A >标记符号之前要使用<Img>标记，图像链接的标记格式如下：

```
<A Href="URL"><Img Src="Picname"></A>
```

其中，URL 是链接目标的 URL，Picname 是图像文件的 URL。在网页内插入图像链接时，浏览器窗口的图像周围将出现黑色边框，如果用户不希望出现该边框，可在<Img>标记符号内添加<Border=0>。

2. 设置网页图像的格式与布局

在默认的情况下，将图像插入到网页文件之后，它与网页中的文本是垂直居下对齐的，并且文本出现在图像的右侧 。要对图像进一步设置，需要了解更多的属性设置方法，如表 2-2 所示。

表 2-2　图像的属性及其用途

属　　性	用　　途
Img Src=" "	图片来源
Img Width=" " Height=" "	图片大小，此宽度及高度一般采用像素作单位。通常设为图片的真实大小，以免失真，若需要改图片大小最好使用专用的图像编辑工具
Img hspace=" "vspace=" "	设定图片边沿空白，以免文字或其他图片贴近。hspace 用于设定图片左右的空间，vspace 用于设定图片上下的空间，高度采用像素作单位
Border=" "	图片边框厚度
Align="top"	调整图片旁边文字的位置，可选值有 top、middle、bottom、left、Right，默认值为 bottom
alt=" "	用以描述该图形的文字，若使用的浏览器不能显示该图片时，这些文字将会代替图片被显示。若浏览器显示了该图片，当鼠标移至图片上该文字也会显示
lowsrc=" "	设定先显示低解析度的图片。若在网页中加入的是一张很大的图片，用户浏览时可能需要很长的下载时间。而设置一张低解析度的图片后，它会先被显示以免浏览者失去兴趣，通常采用原图的黑白版本作为低解析度图片

【练习 2-4】参考上面所介绍的内容，创建如图 2-7 所示包含各种图片格式的网页。

(1) 创建一个文件夹，然后将图 2-8 所示的 5 个图片文件(1.jpg、2.jpg、3.jpg、4.jpg 和 5.jpg)复制至该文件夹中。

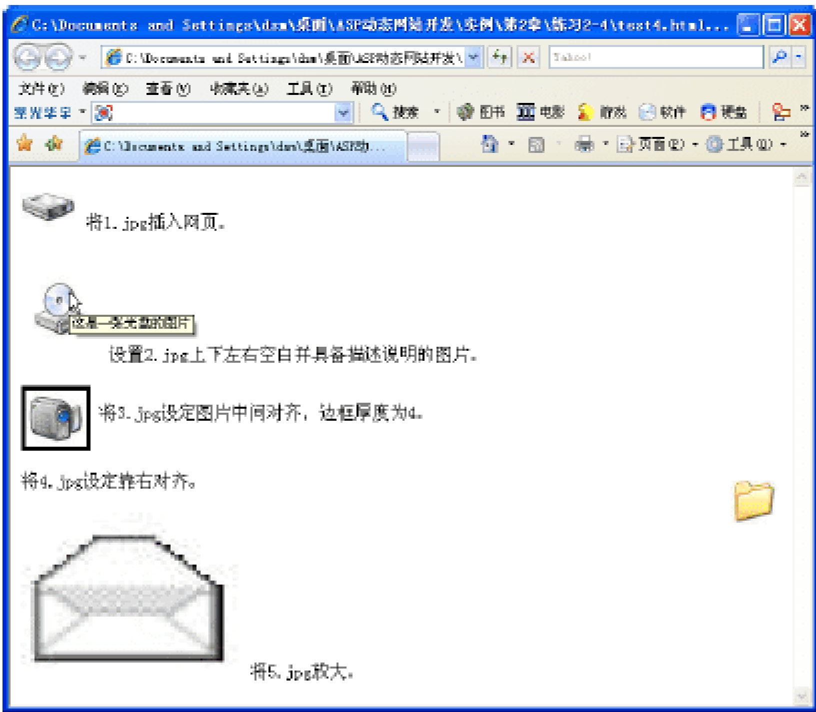


图 2-7　test4.html 的效果



图 2-8　复制图片

(2) 创建一个名为 test4.html 的网页，其代码如下所示(如图 2-9 所示):

```
<p><Img Src="1.jpg" > 将 1.jpg 插入网页。</p>
<p><Img Src="2.jpg" Alt="这是一张光盘的图片" hspace=10 vspace=20> 设置 2.jpg 上下左右空
    白并具备描述说明的图片。</p>
<p><Img Src="3.jpg"  Border=4 Align="Middle"> 将 3.jpg 设定图片中间对齐，边框厚度为 4。
  </p>
<p><Img Src="4.jpg"  Align="Right" > 将 4.jpg 设定靠右对齐。</p>
<p><Img Src="5.jpg" Width=200 Height=150 > 将 5.jpg 放大。</p>
```

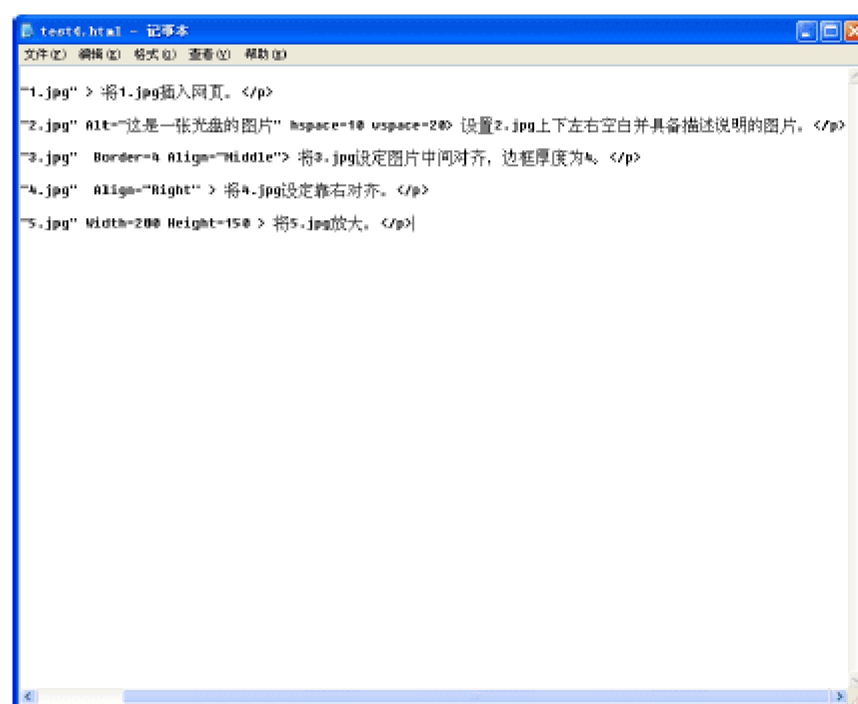


图 2-9　test4.html 的代码

(3) 运行 test4.html 网页，效果如图 2-7 所示。

注意：

【练习 2-4】的代码中所提到的图片文件，必须预先存放在该 HTML 文档所在的文件夹中，才能在浏览器中正常显示。

3. 在页面中加入水平线

使用<Hr>标记可以在网页中插入一条水平线，它的使用方式如下：

```
<Hr Align=对齐方式 Width=x%,Size=n,Noshade>
```

<Hr>标记具有 Size、Color、Width、Align 和 Noshade 属性，各属性的含义如下。

- Size 属性：用于设置水平线的厚度，默认单位是像素。
- Width 属性：用于设置水平线宽度，默认单位是像素，也可使用占浏览器窗口的百

分比来设定。

- Color 属性：用于设置颜色。
- Align 属性：用于设置水平线的对齐方式。
- Noshade 属性：不用赋值，直接加入即可使用，它用来加入一条没有阴影的水平线(不加入此属性水平线将有阴影)。

【练习 2-5】参考上面所介绍的内容，创建如图 2-10 所示的包含各种水平线的网页。

(1) 创建一个名为 test5.html 的网页，其代码如下所示(如图 2-11 所示)：

```
直接插入的水平线<Hr>
宽度为屏幕一半，居中对齐的水平线<Hr Align=Middle Width=50%>
宽度为 300 像素，靠右对齐的水平线<Hr Align=Right Width=300>
厚度为 5 像素的蓝色水平线<Hr Width=50% Size=5 Color="#0000FF">
厚度为 10 像素的无阴影水平实线<Hr Width=50% Size=10  Noshade>
```

(2) 运行 test5.html 网页，效果如图 2-10 所示。

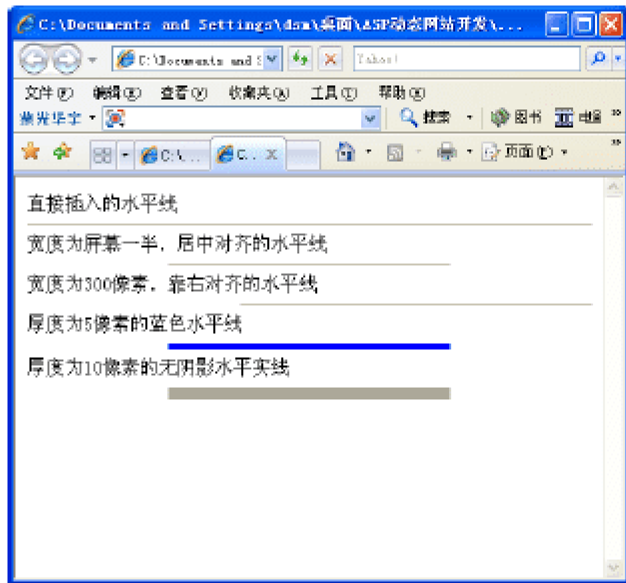


图 2-10　test5.html 的效果

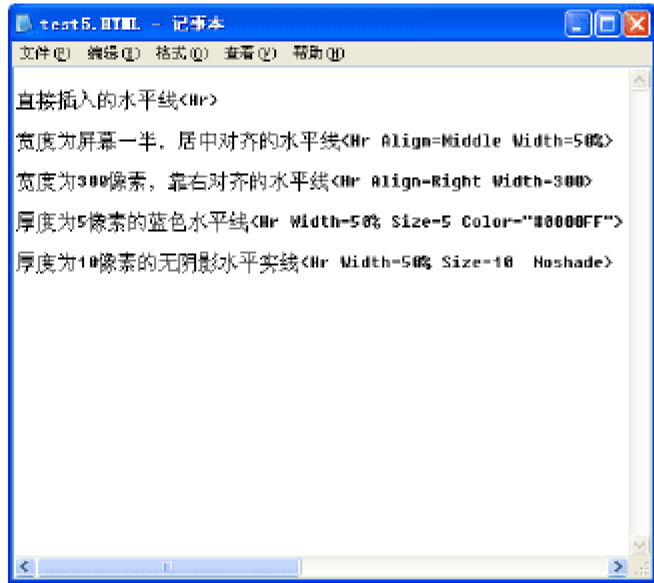


图 2-11　test5.html 的代码

2.2.3　建立超链接

超链接一般可分为外部链接(External Link)与内部链接(Internal Link)。单击外部链接时，浏览器窗口将显示其他文档的内容；单击内部链接时，访问者将看到网页的其他部分并显示在当前浏览器窗口中。另外，HTML 还可以创建指向邮件地址的链接，单击该链接后便可通过电子邮件软件给指定的地址发送邮件。

1. 创建外部链接

一个超链接通常由以下 3 部分构成：首先是超链接标记<A></A>，表示这是一个链接；然后是属性 Href 及其值，这就定义了超链接所指的目标；最后是在超链接中显示在网页上作为链接的文字。链接文本的格式如下：

```
<A Href="URL">
```

单击外部链接时，可在访问者的浏览器窗口中打开并跟踪其他文档。文档可能保存在其他站点内，也可能保存在当前站点内。为了便于区分这两种情况，可将指向其他站点文

档的链接称为 URL 链接，而将指向同一站点内文档的链接称为本地链接。

创建 URL 链接时，需要给出 URL 链接的完整网址，例如下面的代码就是在网页中添加“新浪网”的超级链接：

```
<A Href="http://www.sina.com.cn">新浪网</A>
```

在一台计算机上对不同文件进行链接叫本地链接，常用相对路径或绝对路径表示一个文件。假如链接的目标位于当前 HTML 文档所在文件夹的子文件中，则可直接使用相对路径地址来指向该目标。例如：

```
<A Href="bbs\login.html">登录 BBS</A>
```

若使用绝对路径地址来指向目标，则可表示为：

```
<A Href="/c:\myweb\bbs\login.html">登录 BBS</A>
```

注意：

“/”表示当前路径地址为绝对路径。假如链接的目标位于当前 HTML 文档所在文件夹的上层文件夹中，则路径使用“../”来指向上层文件夹。

此外，<A Href=""></A>还具有 Target 属性，此属性用来指明浏览时的目标框架，该属性各项值的用法与含义如表 2-3 所示。

表 2-3　Target 属性的取值与用途

属　性	用　途
Target="框架名称"	这只运用于框架网页中，若设定则目标网页将显示在“框架名称”的框架中，框架名称是事先由框架标记命名
Target="_blank" 或 Target="new"	将链接目标的内容在新的浏览器窗口中打开
Target="_parent"	将链接目标的内容作为上一个页面
Target="_self"	将链接目标的内容显示在当前窗口中(默认值)
Target="_top"	将框架中链接目标的内容显示在没有框架的窗口中(即除去了框架)

如果不使用 Target 属性，当浏览者单击了链接之后将在原来的浏览器窗口中浏览新的 HTML 文档。若 Target 的值等于_blank，单击链接后将会打开一个新的浏览器窗口来浏览新的 HTML 文档，如下面代码所示：

```
<A Href="http://www.163.com"  Target="_blank">网易网站</A>
```

2. 创建内部链接

所谓内部链接就是网页中的书签。在内容较多的网页内建立内部链接时，它的链接目标不是其他文档，而是网页内的其他位置。在使用内部链接之前，需要在网页内确定书签的位置，并使用<A>标记的 Name 属性为书签命名。内部链接的一般格式为：

```
<A Name="书签名称">书签内容</A>
```

其中，“书签名称”是代表“书签内容”的字符串，用户可使用简短、有意义的字符串代替网页文本。为了使 Web 浏览器易于区分“书签名称”与文档内容，“书签名称”前需要添加符号#。例如，先定义一个标签 A，然后找到“标签名”标签，就可编写如下代码：

```
<A name="标签 A ">书签内容</A>
<A Href="#标签 A ">单击此处将使浏览器跳到“标签 A ”处</A>
```

3. 创建邮件链接

邮件链接可使访问者在浏览页面时，只需单击电子邮件链接就能够打开默认的邮件编辑软件，向指定的地址发送邮件。电子邮件链接的应用格式如下：

```
<A Href="mailto:E-mail 地址"> 邮件链接文本</A>
```

其中，“E-mail 地址”是用户在 Internet 上的电子邮件地址，而“邮件链接文本”就是访问者单击的文本。

访问者单击电子邮件链接时，将打开默认的电子邮件编辑软件。例如，使用 Office 系列的 Outlook Express 作为默认的邮件编辑器时，单击电子邮件链接时将打开 Outlook Express 窗口，“收件人”一栏将出现该邮件地址。

2.3　设 置 网 页

本节将通过几个简单的实例，详细介绍通过修改 HTML 网页代码，在网页中设置表格、框架、列表、表单、CSS 样式和动态 HTML 的方法。

2.3.1　在页面中使用表格

表格对于制作网页是很重要的，现在很多网页都是使用多重表格。主要是因为表格不但可以固定文本或图像的输出，而且还可以任意地设置背景和前景颜色。

1. 认识表格标记

表格由<Table>标记开始，</Table>标记结束，表格的内容由<Tr>标记和<Td>标记定义。<Tr>标记说明表格的一个行，表格有多少行就有多少个<Tr>标记；<Td>标记则设定一个单元格来填充表格。

【练习 2-6】创建包含如图 2-12 所示简单表格的网页。

(1) 创建一个名为 test6.html 的网页，其代码如下所示(如图 2-12 所示):

```
<Table   Border=1>
```

```
<Tr>
 <Td>编号</Td>
 <Td>姓名</Td>
 <Td>工资</Td>
</Tr>
<Tr>
 <Td>3201127</Td>
 <Td>杜思明</Td>
 <Td>7000</Td>
</Tr>
</Table>
```

(2) 运行 test6.html 网页，效果如图 2-12 所示。

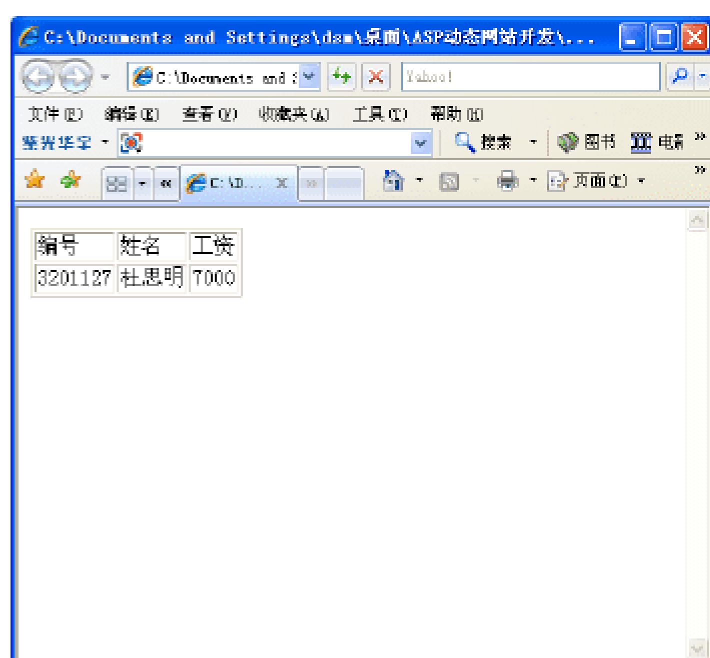


图 2-12　test6.html 的效果



图 2-13　test6.html 的代码

2. 设置表格的整体属性

<Table></Table>标记对用来创建一个表格，它们具有如表 2-4 所示的多种属性。

表 2-4　表格的属性

属　性	用　途
<Table Bgcolor="">	设置表格的背景色
<Table Border="">	设置边框的宽度，若不设置此属性，则边框宽度默认为 0
<Table BorderColor="">	设置边框的颜色
<Table BorderColorlight="">	设置边框明亮部分的颜色(当 Border 的值大于等于 1 时才有用)
<Table BorderColordark="">	设置边框阴影部分的颜色(当 Border 的值大于等于 1 时才有用)
<Table Cellspacing="">	设置表格单元格与单元格之间的空间大小
<Table Cellpadding="">	设置表格单元格边框与其内部内容之间的空间大小
<Table Width="">	设置整个表格的宽度，单位用绝对像素值或总宽度的百分比

注意：

<Table></Table>标记对的各个属性可以结合使用。有关宽度、大小的单位用绝对像素

值，而有关颜色的属性使用十六进制 RGB 颜色码或 HTML 语言给定的颜色常量名。

3. 设置表格单行的属性

<Tr></Tr>标记对用来创建表格中的一行，表格有多少行就有多少对<Tr>标记。<Tr>标记具有如表 2-5 所示的属性。

表 2-5　表格行的属性

属　　性	用　　途
<Tr　Align="">	设置表格行的对齐方式(水平)，可选值为 left、center、right
<Tr　vAlign="">	设置表格行的对齐方式(垂直)，可选值为 top、middle、bottom
<Tr　Bgcolor="">	设置表格行的底色
<Tr　BorderColor="">	设置表格行的边框颜色
<Tr　BorderColorlight="">	设置表格行的边框明亮部分的颜色
<Tr　BorderColordark="">	设置表格行的边框阴影部分的颜色

【练习 2-7】通过对表格行的属性设置，创建如图 2-14 所示的表格网页。

(1) 创建一个名为 test7.html 的网页，其代码如下所示(如图 2-15 所示):

```
<Table Width="85%" Border="1" Cellspacing="5"  BorderColor="black">
<Tr BorderColor="#0000FF" Align="Right">
<Td>第一行边界线为蓝色</Td><Td>第一行靠右对齐</Td>
    </Tr>
    <Tr BorderColorlight="#CF0000"  BorderColordark="#00FF00"  vAlign="bottom">
<Td>第二行向光边框为绿色背光边框为红色</Td><Td>第二行靠底对齐</Td>
    </Tr>
</Table>
```

(2) 运行 test7.html 网页，效果如图 2-14 所示。

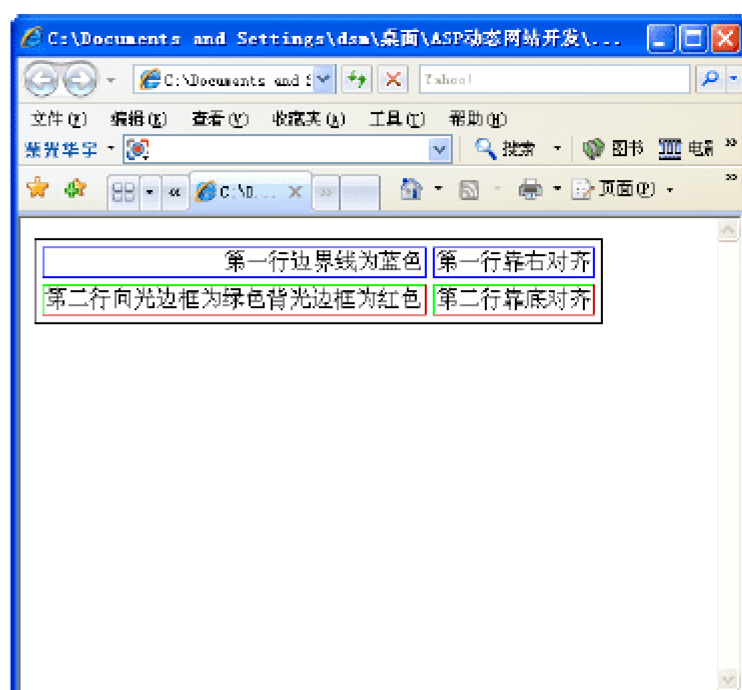


图 2-14　test7.html 的效果

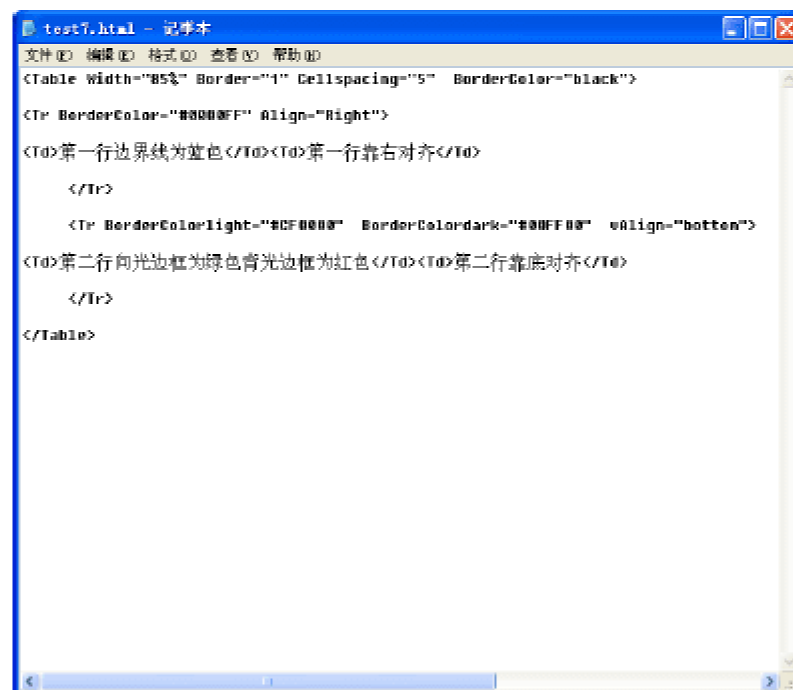


图 2-15　test7.html 的代码

4. 设置普通单元格属性

<Td></Td>标记对用来设置表格中的一个单元格的内容及格式。单元格中可以包含文本、图像、列表、段落、表单、水平线、表格等。<Td>标记具有如表 2-6 所示的属性。

表 2-6　单元格的属性

属　性	用　途
<Td　Width="">	设置单元格的宽度，接受绝对值(如 80)及相对值(如 80%)
<Td　Height="">	设置单元格的高度
<Td　Colspan="">	设置单元格向右合并的单元格数
<Td　Rowspan="">	设置单元格向下合并的单元格数
<Td　Align="">	设置单元格的对齐方式(水平)，可选值为 left、center、right
<Td　vAlign="">	设置单元格的对齐方式(垂直)，可选值为 top、middle、bottom
<Td　Bgcolor="">	设置单元格的底色
<Td　BorderColor="">	设置单元格的边框颜色
<Td　BorderColorlight="">	设置单元格的边框明亮部分的颜色
<Td　BorderColordark="">	设置单元格的边框阴影部分的颜色
<Td　Background="">	设置单元格的背景图片，与 Bgcolor 任选其一

5. 设置标题单元格属性

<Th>标记与<Td>标记同样是标记一个单元格，唯一不同的是<Th>标记所标记的单元格中的文字以粗体出现，通常用于表格中的标题栏。用它取代<Td>标记的位置就可以了，其属性设定请参考<Td>标记。为<Td>标记所标记的文字加上粗体标记也能达到同样的效果。

6. 设置表格的总标题

<Caption>标记的作用是为表格加上一个标题，如同在表格上方加上没有格线的通栏行，通常用来存放表格标题。

可使用<Caption Align=" " >属性来设置表格标题行相对于表格的对齐方式(水平)，可选值为 left、center、right、top、middle 与 bottom。若 Align="bottom"，标题列便会出现在表格的下方，而与<Caption>标记语句在<Table>标记中的位置无关。

【练习 2-8】通过对单元格的属性设置，创建如图 2-16 所示的表格网页。

(1) 创建一个名为 test8.html 的网页，其代码如下所示(如图 2-17 所示):

```
<Table Width="350" Border="1" Cellspacing="0" Cellpadding="2" Align="center"
    Bgcolor="#FFC4E1" BorderColor="#0000FF">
<Caption>格式单词与其含义</Caption>
     <Tr Align="center">
<Td colspan="3">横向通栏示例</Td>
```

```
</Tr>
        <Tr Align="center">
<Td rowspan="3">纵向通栏示例</Td>
<Th>格式单词</Th><Th>含义</Th>
        </Tr>
        <Tr Align="center">
<Td>Width</Td><Td>宽度</Td>
        </Tr>
        <Tr Align="center">
<Td>Height</Td><Td>高度</Td>
        </Tr>
</Table>
```

(2) 运行 test8.html 网页，效果如图 2-16 所示。

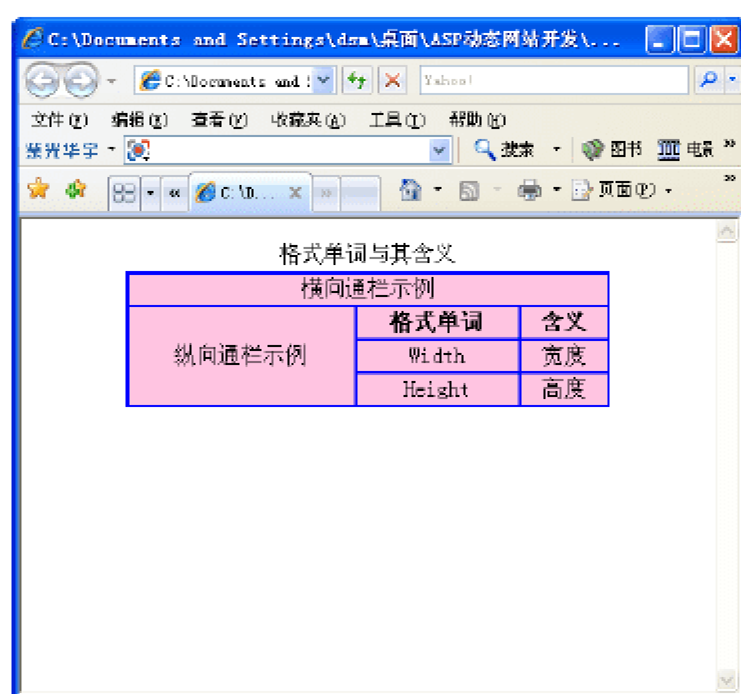


图 2-16　test8.html 的效果

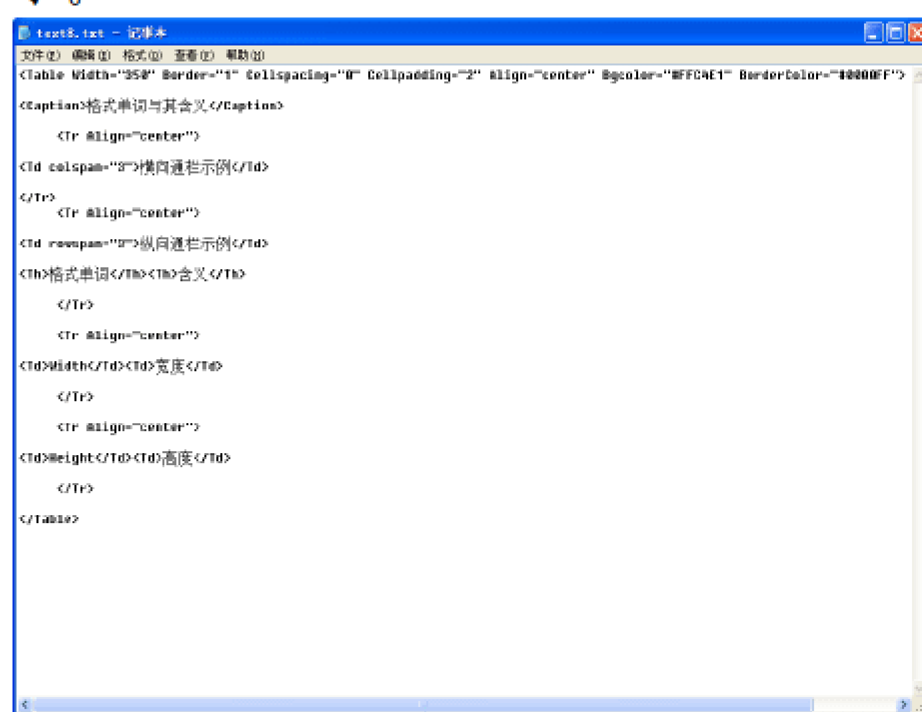


图 2-17　test8.html 的代码

2.3.2　在页面中使用框架

框架网页把浏览器窗口切割成几个独立的部分，打开的链接目标文件只占用浏览器窗口的某个区域，该区域就是框架网页的目标框架。框架网页的出现，使得访问者在浏览器窗口中可同时观察多个网页。

1. 认识框架标记

设计框架网页时，<Frame>标记和<Frameset>标记用于定义框架网页的结构。由于框架网页的出现，从根本上改变了 HTML 文档的传统结构，因此在出现<Frameset>标记的文档中，将不再使用<Body>标记，包含框架网页的 HTML 文档的基本结构为：

```
<Html>
<Head>…</Head>
<Frameset>…</Frameset>
<Frame Src="URL">
</Html>
```

其中，URL 用于确定在框架网页内显示的网页文件的地址。

注意：

如果考虑到一些不支持框架网页功能的浏览器，可使用<Noframes></Noframes>标记对，把此标记对放在<Frameset></Frameset>标记对之间。

在网页内添加框架网页，就意味着对浏览器窗口进行纵向与横向的划分。Rows 用来规定主文档中各个横向划分的框架的行定位，而 Cols 用来规定主文档中各个纵向划分的框架的列定位。这两个属性的取值可以是百分数、绝对像素值或星号“*”，其中星号代表那些未被划分的空间，如果同一个属性中出现多个星号则将剩下的未被说明的空间平均分配。同时，所有的框架将按照 Rows 和 Cols 的值从左到右，然后从上到下排列。设置框架网页大小尺寸的例子如下所示：

```
<Frameset Rows="*,*,*">
```

该例共设置 3 个按列排列的框架，每个框架占整个浏览器窗口的 1/3。

```
<Frameset Cols="40%,*,*">
```

该例共设置 3 个按行排列的框架，第一个框架占整个浏览器窗口的 40%，剩下的空间平均分配给另外两个框架。

```
<Frameset Rows="40%,*" Cols="50%,*,200">
```

该例共设置 6 个框架，先是在第一行中从左到右排列 3 个框架，然后在第二行中从左到右再排列 3 个框架，即两行三列，所占空间依据 Rows 和 Cols 属性的值，其中 200 表示 200 像素。

注意：

使用<Frameset>标记时，Rows 和 Cols 这两个属性至少选择一个，否则浏览器只显示第一个定义的框架。如果要固定框架的结构大小，不允许用户在浏览时拖动改变框架的大小，可在 HTML 代码中添加一句<Frame NoreSize>。

2. 确立框架目标

在框架网页内单击超链接后，链接目标就会出现在目标框架内。在确定目标框架之前，应该为它命名，通过框架网页的名称来确定目标框架的位置，框架网页的名称应该注意区分大小写。内容相同，大小写不同的框架网页名称将被认为是不同的框架网页。确定目标框架网页的通用格式如下：

```
<Frame Name="框架网页名称">
<A Href=URL,Target="框架网页名称">
```

对于一些特殊的框架网页，HTML 已经预先为其设置了名称，这些常用的特殊框架网页包括如下内容。

- black：空白框架网页。单击链接文本后，将打开一个新的浏览器窗口，并显示链接目标。
- self：将链接指向当前框架网页。单击链接文本后，链接目标将在链接文本所在的框架网页内出现，并且链接文本窗口将被刷新。
- parent：将链接指向父框架网页。如果没有父框架网页，那么它就指向自己。父框架、子框架网页是根据网页的结构关系设置的。
- top：指向整个浏览器窗口本身，它是打开网页时首先看到的浏览器窗口。

3. 设置框架网页的外观

框架网页外观是由框架网页的边框、间距、颜色、页边距、滚动条等组成的。默认情况下，HTML 提供了一系列的默认值，分别对上述选项进行设置。根据应用框架网页的背景、场合的不同，用户可以自定义框架网页的外观。

通过设置 FrameBorder 属性，用户可以自定义边框是否出现。设置框架网页边框的应用格式如下所示：

```
<Frame FrameBorder=Yes|No>
```

其中，Yes 表示在浏览器窗口显示框架网页边框，如果用户选择 No，框架网页边框将消失。类似地，将 FrameBorder 属性设置为 0 时，框架网页边框也将消失，但设置任何大于 0 的数值时，框架网页边框都会出现，并且宽度是一致的。

框架网页间距是指框架网页之间的空白区域，框架网页的内容不会出现在该区域。使用<Frameset>标记的 Framespacing 属性可以设置不同的框架网页间距，当需要将浏览器框架网页内所有的框架网页间距设置为 50 像素时，可在网页文件内添加下列语句：

```
<Framest Cols="10%,*" Framespacing=50>
```

每个框架网页都相当于一个独立的网页，因此可对网页的页边距进行设置。<Frame>标记的 MarginLength 和 MarginHeight 属性分别用于设置页边距的宽度和高度。

当框架网页的内容超过框架网页的大小尺寸时，可以使用滚动条拖动的方式来观察整个网页的内容。用户可通过 Frame 标记的 Scrolling 属性决定是否允许滚动条出现在浏览器窗口。用户可将 Yes、No、Auto 赋值给 Scrolling 属性。默认情况下，系统将给 Scrolling 属性赋值为 Auto，这样可根据框架网页内容的多少，决定是否在浏览器窗口内出现滚动条。

【练习 2-9】创建如图 2-18 所示的框架网页。

(1) 将计算机接入 Internet，并能通过 IE 浏览网页。

(2) 创建一个名为 test9.html 的网页，其代码如下所示(如图 2-19 所示)：

```
<Frameset Cols=20%,*>
```

```
<Frame Src="http://www.BAIDU.com">
    <Frameset Rows=40%,*>
    <Frame Src="http://www.sina.com.cn">
    <Frame Src="http://www.sohu.com.cn">
    </Frameset>
</Frameset>
```

(3) 运行 test9.html 网页，效果如图 2-18 所示。



图 2-18　test9.html 的效果

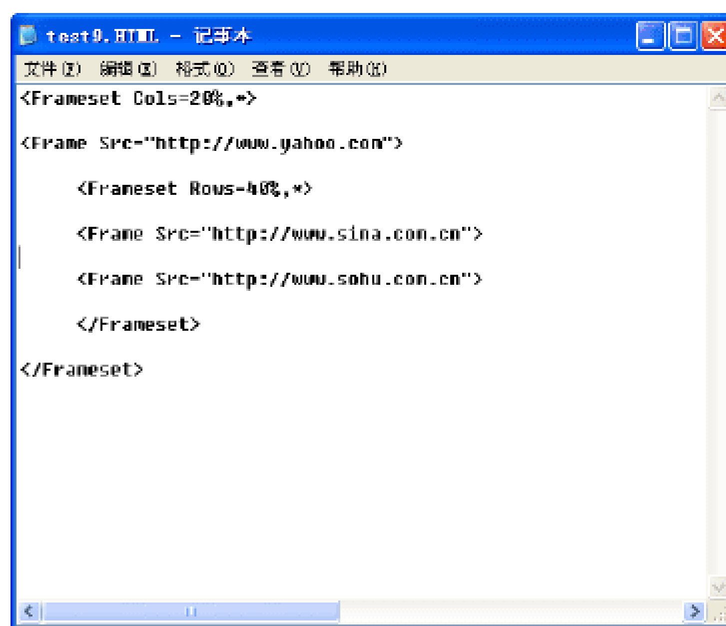


图 2-19　test9.html 的代码

2.3.3　在页面中使用列表

使用列表能够有效地表达出具有并列、排序关系的网页内容，为访问者阅读网页提供方便。HTML 为用户提供了编号列表、符号列表与自定义列表 3 种形式。通过上述列表的相互嵌套，还可以进一步丰富列表的表现方式。

1. 使用编号列表

当网页中的某些内容存在排序关系时，可以使用编号列表，以表明这些内容是有前后顺序的。编号列表的应用格式如下：

```
<Ol>
<Li>编号列表
……
</Ol>
```

注意：

在编号列表的开始与结束处，需要使用<Ol></Ol>标记对(Ol 是 Ordered List 英文的缩写)，它用于定义编号列表的作用范围。在编号列表内容之前必须添加<Li>列表项标记(它是列表选项 List Item 英文的缩写)，以便与其他列表相区别。

编号列表支持 Type 属性，属性值与编号类型的对应关系如表 2-7 所示。在默认情况下，编号总是从该类型的第一个数值或字母开始，通过 Type 属性用户可以设置编号列表不同

的起始序号。

表 2-7　<Li>标记的 Type 属性设置

Type 属性	编号显示方式
A	英文大写字母，如 A、B、C 等
A	英文小写字母，如 a、b、c 等
I	罗马大写字母，如Ⅰ、Ⅱ、Ⅲ等
I	罗马小写字母，如ⅰ、ⅱ、ⅲ等
L	阿拉伯字母，如 1、2、3 等

2. 使用符号列表

当网页内容出现并列选项时，可采用符号列表。它的标记是<Ul>(UI 是 Unordered List 英文的缩写)，在每一列表项的开始处须使用<Li>标记以示区别。符号列表的使用格式如下：

```
<Ul>
<Li>符号列表
……
</Ul>
```

默认情况下，符号列表的项目符号是圆点，改变 Type 属性的赋值时，可以更换项目符号的形式，用户可在 Disc(圆点)、Circle(圆圈)、Square(方块)中选择满意的项目符号。将 Type 属性值添加到<Ul>标记内，所有的列表项目都采用相同的符号项目。将 Type 属性值添加到<Li>标记内，它只能改变当前列表的项目符号，通过这种方法可为列表内的项目设置不同的项目符号。

3. 自定义列表

当网页内出现新词汇、术语时，为了给访问者一个明确的提示，需要对它们进行定义和说明，此时用户可以使用自定义列表(Definition List)。自定义列表标记<Dl>是由一系列的词语标记<Dt>和定义标记<Dd>组成，通常<Dt>标记与<Dd>标记成对出现在网页文件内，词语的定义内容以首行缩进的方式显示在浏览器窗口中。自定义列表的应用格式如下：

```
<Dl>
<Dt>第 1 条词语<Dd>定义内容
<Dt>第 2 条词语<Dd>定义内容.
……
</Dl>
```

4. 定义嵌套列表

HTML 不仅允许用户使用单独的列表，还能够把不同类型的列表相互嵌套。嵌套的级数不受限制，这样就形成复合列表，它意味着第一个列表的内容还未结束时，另一列表就

可以开始。例如，在自定义列表内，使用编号列表说明具有层次感的列表选项，使用符号列表说明具有并列关系的列表选项。

【练习 2-10】创建如图 1-20 所示的包含各种列表的网页。

(1) 创建一个名为 test10.html 的网页，其代码如下所示(如图 2-21 所示)：

```
<Dt>
<Dt>ordered<Dd>现代英汉词典解释
<Ol>
<Li>安排好的；整齐的；<Li>规则的；<Li>有秩序的；
</Ol>
<Dt>HTML 列表<Dd>列表的种类
<Ul>
<Li>编号列表<Li>符号列表<Li>自定义列表
</Ul>
</Dl>
```

(2) 运行 test10.html 网页，效果如图 2-20 所示。

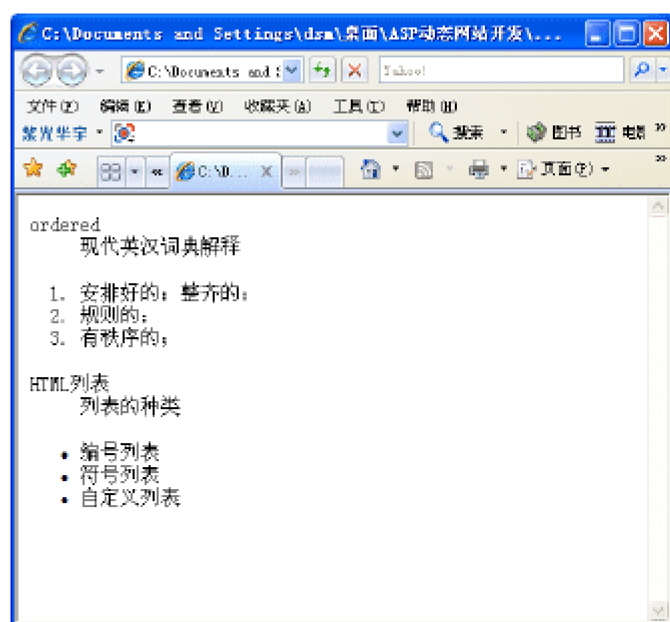


图 2-20　test10.html 的效果

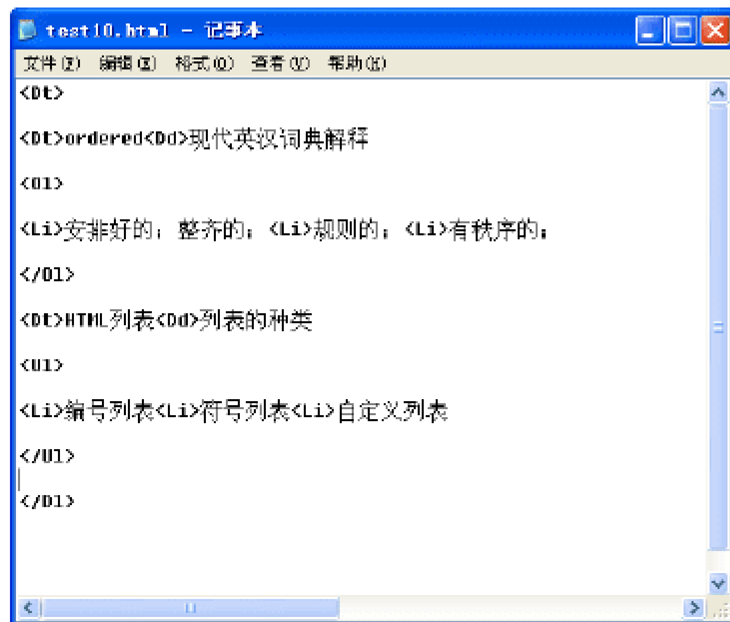


图 2-21　test10.html 的代码

2.3.4　在页面中使用表单

在 Web 网页中，表单用来给供访问者填写信息，从而使管理员能获得访问者信息，使网页具有交互功能。表单设计在 HTML 文档中，当用户填写完信息后做提交(submit)操作，于是表单的内容就从客户端的浏览器传送到服务器上，经过服务器上的 ASP 或 PHP 等处理程序处理后，再将用户所需信息返回到客户端的浏览器上，这样网页就具有了交互性。本节将从最基本的表单元素开始，介绍如何使用 HTML 的表单标记来设置表单。

1. 表单的基本结构

网页内的表单由表单标记<Form>定义，使用<Form>标记意味着表单的开始，而</Form>标记符号则意味着表单的结束。由于表单通常用于收集站点访问者的信息，因此在表单的内部必须出现输入标记<Input>，用于收集表单数据。另一方面，还可将表单数据

发送给站点管理员，或清除表单的内容，重新输入表单。表单标记的基本结构如下所示：

```
<Form Action=URL Method=get|post>
…
<Input Type=Submit>
<Input Type=reset>
</Form>
```

表单标记<Form>最重要的属性就是 Action 和 Method。Action 属性用于指定表单处理程序的 URL。例如<form action="login.asp">，当用户提交表单时，服务器将执行该 HTML 文件所在文件夹中名为 login.asp 的 ASP 程序。

Method 属性用于定义处理站点访问者提供数据的方法，可取值为 GET 或 POST。在 GET 方式下，处理程序从当前 HTML 文档中获取数据，然而这种方式传送的数据量是有所限制的，一般限制在 1KB 以下。POST 方式与 GET 方式相反，在 POST 方式下当前 HTML 文档把数据传送给处理程序，传送的数据量要比使用 GET 方式大得多。

2. 表单的用户输入区域

表单是一个能够包含多种不同表单元素的区域。表单元素能够让用户在表单中输入信息，有文本框、密码框、下拉菜单、单选按钮和复选框等。

最常用的表单输入标记是<Input>标记，用来定义输入区，用户可在其中输入信息。<Input Type="">标记提供了 8 种类型的输入区域，由 Type 属性来决定区域类型，如表 2-8 所示。

表 2-8　表单的各项组成元素

Type 属性取值	输入区域类型	输入区域示例
<Input Type="Text" Size="" Maxlength="">	单行文本输入区域。Size 与 Maxlength 属性用来定义显示的大小与可输入的最大字符数	姓名：
<Input Type="Submit">	将表单内容提交给服务器的按钮	提交查询内容
<Input Type="Reset">	将表单内容全部清除，重新填写的按钮	重置
<Input Type="Checkbox Checked>	复选框。Checked 属性用来设置复选框在默认情况下是否被选中	请选择你的爱好 ☑ 音乐 ☑ 体育 ☑ 文学
<Input Type="Hidden">	隐藏区域。用户不能在其中输入，常用来预设某些要传送的信息	—
<Input Type="Image" Src="URL">	使用图像来代替 Submit 按钮，图像的源文件名由 Src 属性指定。用户单击后，表单中的信息和单击位置的 X、Y 坐标一起传送给服务器	—
<Input Type="Password">	输入密码的区域。当用户输入密码时，区域内将会显示“*”号	请输入密码：
<Input Type="Radio" Checked>	单选按钮。Checked 属性用来设置单选按钮默认情况下是否被选中	请输入性别： ○ 男 ○ 女

上面 8 种类型的输入区域有一个公共的属性 Name，此属性为每一个输入区域设置一个名字，一个输入区域对应一个名字，服务器就是通过调用某一输入区域的名字的 Value 属性来获得该区域的数据，而 Value 属性是另一个公共属性，它可用来指定输入区域的默认值。

3. 表单的列表框

列表框是确定选项内容的另一种方式，它包括下拉列表框和滚动列表框两种，在下拉列表框内，只能选择其中的一个选项；在滚动列表框内，则可以选择其中的多项内容。表单的列表框是由<Select>和<Option>两个标记来定义的，它的使用格式如下所示：

```
<Select Name= "name">
<Option>
</Select>
```

<Select>标记具有 Multiple、Name 和 Size 等属性。Multiple 属性无需赋值，直接加入标记中即可使用，加入此属性后列表框就成了多选的了；Name 属性用于确定该控件的名称；Size 属性用来设置列表的高度，默认值为 1。

<Option>标记用来指定列表框中的一个选项，它放在<Select></Select>标记对之间，此标记具有 Selected 和 Value 属性。Selected 属性用来指定默认的选项；Value 属性用来给<Option>标记指定选项赋值，这个值是要传送到服务器上的，服务器正是通过调用<Select>区域的名字的 Value 属性来获得该区域选中的数据项。

【练习 2-11】创建如图 2-22 所示的包含列表框的表单网页。

(1) 创建一个名为 test11.html 的网页，其代码如下所示(如图 2-23 所示):

```
<form action="apply.asp" method="post">
  <P>请选择你的年龄:
  <Select name="old" Size="1">
    <option Value="少年">9~18
    <option Value="青年" selected>19~35
    <option Value="中年">36~55
    <option Value="老年">55 以上
  </Select>
  <P>请选择你所在的城市:
  <Select name="city" multiple Size="4">
    <option Value="beijing">北京
    <option Value="shanghai" selected>上海
    <option Value="nanjing">南京
    <option Value="zhengzhou">郑州
  </Select>
</form>
```

(2) 运行 test11.html 网页，效果如图 2-22 所示。

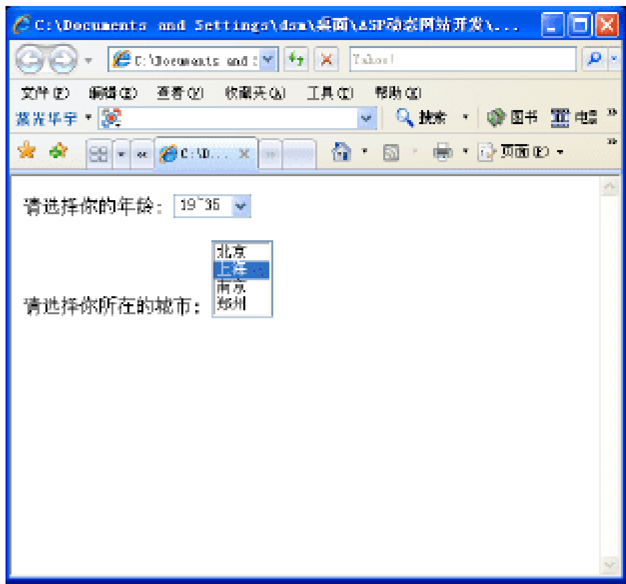


图 2-22　test11.html 的效果

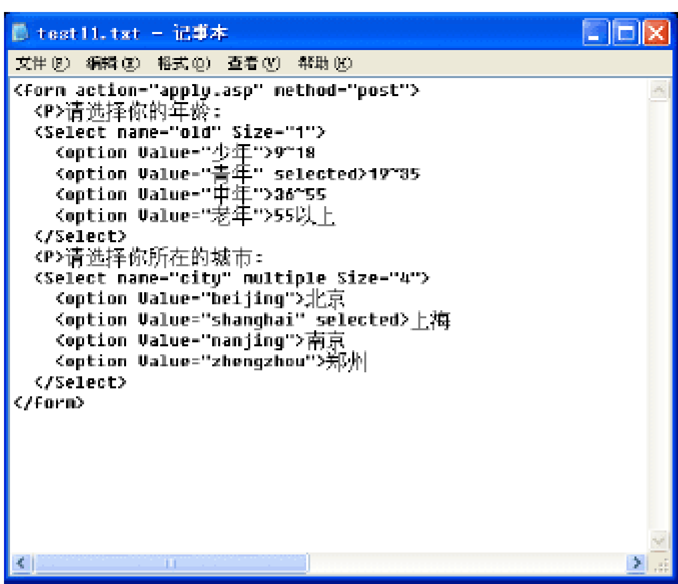


图 2-23　test11.html 的代码

4. 文本框与文件选项

<Textarea></Textarea>标记对用来创建一个可以输入多行文本的文本框，此标记对用于<Form></Form>标记对之间。<Textarea>标记时具有 Name、Cols 和 Rows 属性。Cols 和 Rows 属性分别用来设置文本框的列数和行数，这里列与行是以字符数为单位的。

如果在表单内填写的内容太多，例如个人工作经历等，为了方便访问者填写，可在表单内添加文件选项。

在表单内添加文件选项时，用户可使用<Form>标记的 Enctype 属性来指定文件的数据类型，使用该属性还需要将<Input>标记的 Type 属性设置为 File。

【练习 2-12】创建如图 2-24 所示的包含文本框与文件选项的表单网页。

(1) 创建一个名为 test12.html 的网页，其代码如下所示(如图 2-25 所示):

```
<Form Action="select.asp"Method="post">
  <P>请输入留言:
  <Textarea name="ly" Cols="20" Rows="4">
请在这里输入您的看法和见解
  </Textarea>
<Br><Br><Hr Align=left><Br>
请选择上传的文件:<Input name="filename" Type="file"><P>
<Input Type=Submit Value="提交">
<Input Type=Reset Value="重选">
</Form>
```

(2) 将输入的代码保存后，测试网页，效果如图 2-24 所示。

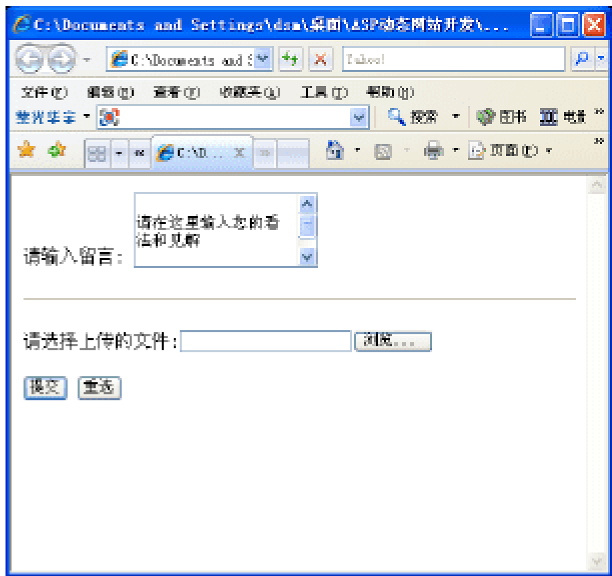


图 2-24　test12.html 的效果

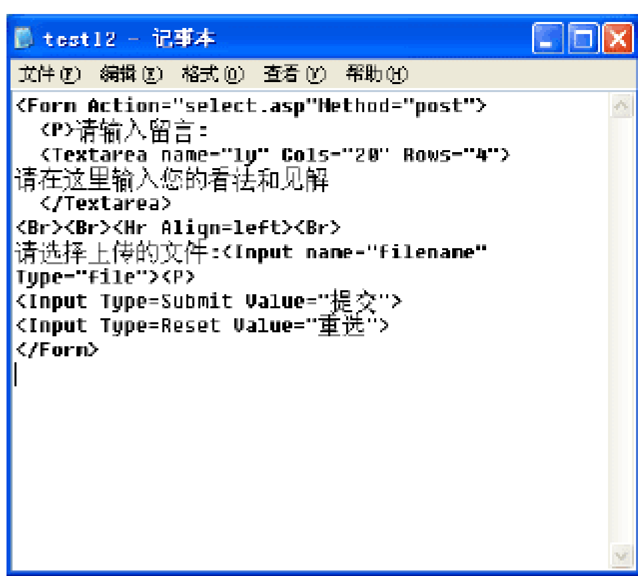


图 2-25　test12.html 的代码

2.3.5　使用动态 HTML(DHTML)

使用 DHTML 技术，可通过浏览器技术、客户端脚本技术和 CSS 样式表生成变化缤纷的动态页面。

DHTML 是 Dynamic HTML(动态 HTML)的简称，它是一种即使在脱离网络环境的情况下，网页装载到浏览器后仍然能够随时变换内容的技术。例如，当鼠标指针移至文章段落中，段落能变成醒目的红色或蓝色；在网页的页面上，随机地漂浮着一些漂亮的小标志；网页横幅不断地滑动或具有滚动闪亮的效果等。

DHTML 之所以能够制作出“动感”的效果，首先是因为 DHTML 将网页中的每个元素划分成了许多独立的对象，通过 CSS 来指定这些对象的属性。

CSS 是 DHTML 网页进行改变的对象，网页中的各种变化都是通过脚本语言设置 CSS 的属性来实现的。因此可以说 CSS 是编写 DHTML 网页的基础。

CSS 本身是 HTML 的扩展，因此语法规则并不复杂，掌握 HTML 语言使用方法后就不难看懂 CSS 代码。CSS 的功能十分强大，它能够对文字间距、字体、列表、颜色、背景、位置等多种属性进行精确控制。

将 CSS 样式表添加到 HTML 文档的方法有多种，可链接外部样式表、嵌入样式表，以及输入样式、内联样式等。

另外，CSS 还有具有艺术功能的滤镜效果。滤镜效果将可视化的滤镜和转换效果添加到标准的 HTML 元素上，可进行有序地变化和更替演示，产生精彩的渐变和动画效果。运用好动态网页技术，可使制作的网页美仑美奂，大放异彩。

2.4　习　　题

2.4.1　填空题

(1) HTTP 即___________协议，它是 WWW 服务器使用的最主要的协议。

(2) ___________标记用于 HTML 文档的最前边，用来标识 HTML 文档的开始。而___________放在 HTML 文档的最后边，用来标识 HTML 文档的结束。

(3) <Hr>标记的 Width 属性用于设置水平线宽度，默认单位是___________，也可使用占浏览器窗口的百分比来设定。

(4) 在<Body>标记的属性中，用于设置网页背景颜色的是___________，用于设置网页文本颜色的是___________。

(5) 表单标记<Form>的___________属性用于指定表单处理程序的 URL，___________属性用于定义处理站点访问者提供数据的方法。

(6) <Input Type="">标记的___________属性用于为输入区域命名，___________属性用来指定输入区域的默认值。

2.4.2　选择题

(1) 下面关于标记的说法不正确的是(　　)。

A. 标记需要填写在一对尖括号(< >)内

B. 在书写标记时，英文字母的大小写或混合使用大小写都是允许的

C. 标记内可以包含一些属性，属性名称出现在标记的后面，并且以分号进行分隔

D. HTML 对属性名称的排列顺序没有特别的要求

(2) 以下标记中，(　　)可用于在网页中插入图像。

A. <Img>标记　　B.
标记

C. <H3>标记　　D. <src>标记

(3) 下面关于表格属性的说法不正确的是(　　)。

A. Border 值为边框的宽度，若不设置此属性，则边框宽度默认为 0

B. Cellspacing 属性用于设置表格单元格与单元格之间的空间大小

C. Width 属性用于设置整个表格的宽度，单位用绝对像素值或总宽度的百分比

D. Color 属性用于设置表格的颜色，可用 6 个十六进制数来表示

(4) 在超链接中如果指定(　　)框架名称，链接目标将在链接文本所在的框架网页内出现，并且链接文本窗口将被刷新。

A. Black　　B. Self　　C. Parent　　D. Top

(5) 用户可通过<Frame>标记的(　　)属性决定是否允许滚动条出现在浏览器窗口。

A. FrameBorder　　B. Scrolling

C. Name　　D. MarginHeight

(6) <Input　Type="Reset">是一个(　　)。

A. 文本框　　B. 重新填写的按钮

C. 下拉菜单　　D. 提交给服务器的按钮

2.4.3　操作题

(1) 利用各种编辑文本的 HTML 标记建立如图 2-26 所示的网页。

(2) 利用建立表单的各种 HTML 标记建立如图 2-27 所示的网页。

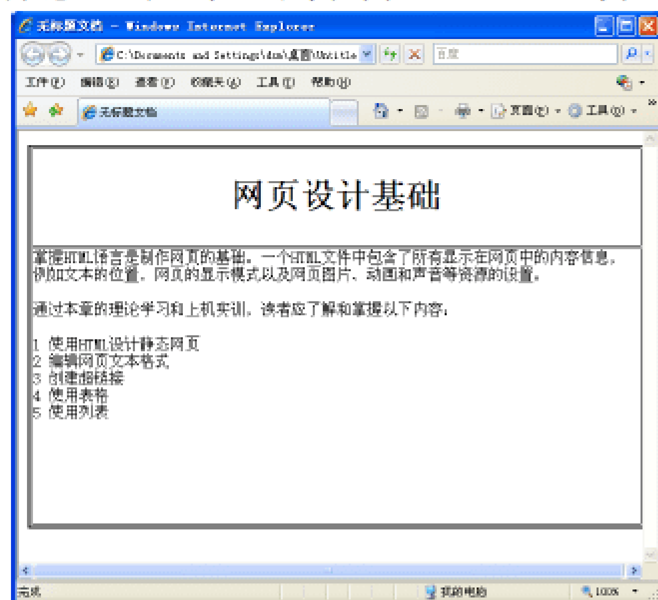


图 2-26　网页效果 1

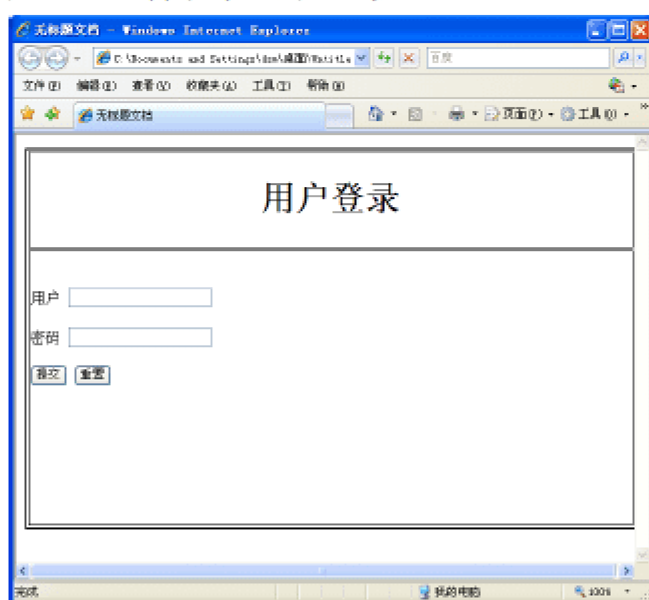


图 2-27　网页效果 2

第3章　VBScript语言基础

VBScript 是 Macromedia 公司开发的一种脚本语言，可以用来编写用于客户端和服务器端上的脚本程序。ASP 利用它创建动态程序代码，用户可以将其视为简化版的 Visual Basic。本章将简要地介绍 VBScript 的应用基础与常用函数，以帮助用户在学习 ASP 时更好地利用其功能。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 了解使用 VBScript 配置 ASP 的方法
- 了解 VBScript 的特点
- 了解 VBScript 的变量
- 了解 VBScript 的常数
- 了解 VBScript 的数据类型
- 掌握 VBScript 常用语句的使用方法
- 了解 VBScript 的过程与函数

3.1　使用 VBScript 配置 ASP

ASP 本身并不是一种脚本语言，但它却为嵌入 HTML 页面中的脚本语言提供了运行环境。在 ASP 程序中常用的脚本语言有 VBScript 和 JScript 等语言，系统默认的语言为 VBScript。在本节和后续章节中，将介绍一些与 ASP 开发相关的或是很重要的 VBScript 基础知识。

脚本是由一系列的脚本命令所组成的，如同一般的程序，脚本可以将一个值赋给一个变量，可以命令 Web 服务器发送一个值到客户浏览器，还可以将一系列命令定义成一个过程。

VBScript 来源于 Visual Basic 的子集 VBA(Visual Basic for Application)。VBA 是 Microsoft Office Suite 和其他应用程序的宏编程语言，利用它可以扩展 Office 的功能。

ASP 能支持多种脚本语言，在编写 ASP 代码时通常有 3 种方法来声明所使用的脚本语言种类：用 IIS 指定默认脚本语言，在 ASP 文件中进行脚本语言声明，以及在<Script>标记中加入所需的语言。

3.1.1　通过 IIS 指定 ASP 使用 VBScript

用户可通过 IIS 指定默认使用的脚本语言，只要是<%和%>之间的代码，ASP 在解释

时会认为使用的是默认脚本语言。

【练习 3-1】通过“Internet 信息服务”控制台来设定 ASP 的默认脚本语言为 VBScript。

(1) 选择“开始”|“运行”命令，打开“运行”对话框，输入 inetmgr 命令，并按 Enter 键。

(2) 系统打开 IIS 的管理工具“Internet 信息服务”控制台，右击欲设置的网站名称，从弹出的快捷菜单中选择“属性”命令，如图 3-1 所示。

(3) 打开“主目录”选项卡，单击“配置”按钮，打开“应用程序配置”对话框，并打开“选项”选项卡，如图 3-2 所示。

(4) 在“默认 ASP 语言”文本框中输入 VBScript，然后单击“确定”按钮。

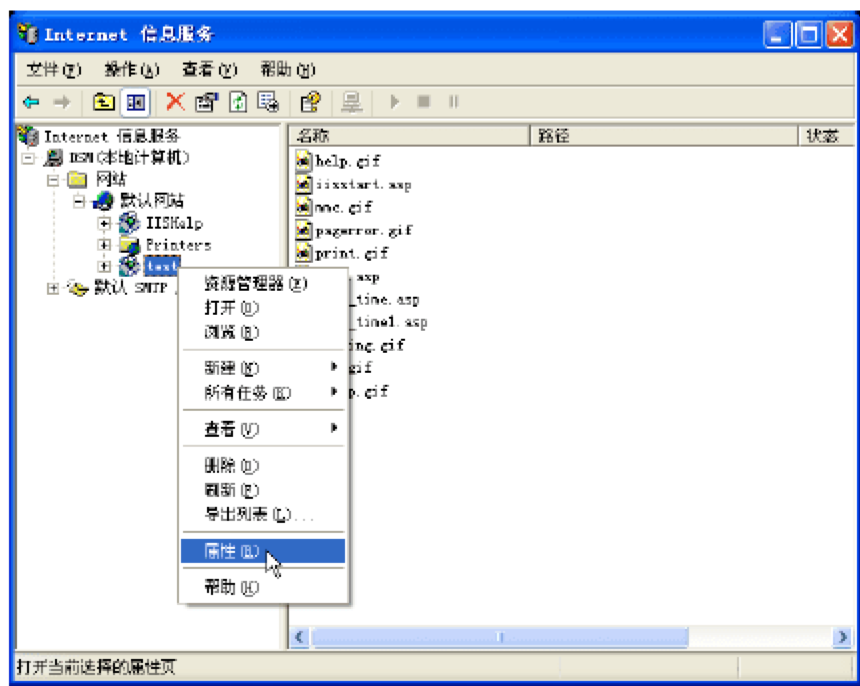


图 3-1　右击需要设置的网站

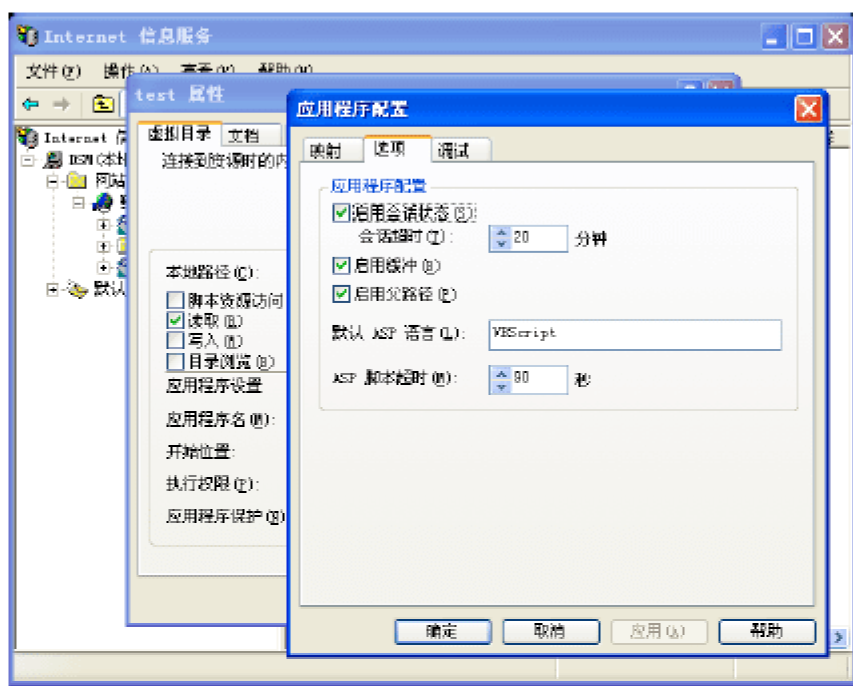


图 3-2　“应用程序配置”对话框

3.1.2　在 ASP 文件中声明脚本语言

如果在某一单个页面指定脚本语言，可在文件初始部分用一条声明语句进行指定。须要特别注意的是，该语句一定要放在所有语句之前，如下例所示：

```
<%@  Language=VBScript %>
<Html><Head>
……
</Html></Head>
```

3.1.3　在<Script>标记中加入脚本语言

最灵活、作用范围最小的脚本语言指定方法是使用<Script>标记，即在<Script>标记中加入所需的语言。在一个已设定主脚本语言的页面中，使用<Script>标记仍可以指定页面的某一部分采用其他的脚本语言，如下例：

```
<Html><Head>
<Body>
<Script Language="VBScript" Runat="Server">
```

```
Response.Write("hello world!")                    '在屏幕上输出 hello world!
</Script>
</Body>
</Html></Head>
```

在这里，<Script>标记的 Language 属性限定了包含的语句使用了哪一种语言。Runat 属性指出该脚本应当在 Server 端还是在 Client 端(浏览器上)实现。有时为了某种需要，可能也会指定 VBScript 代码放在客户端执行，此时将 Runat="Server"删去即可，如下例所示：

```
<Script Language="VBScript" >
Msgbox("这是一个弹出信息对话框")                '弹出一个显示括号内信息的对话框
</Script>
```

注意：

MsgBox 用于显示一个信息对话框，而 InputBox 用于产生一个文本输入框，它们都是解释后立即在当前位置执行的语句，所以它们不能在服务器端脚本中使用。

在 ASP 程序中，可以灵活地反复使用<Script>标记，这样不仅可以混合使用服务器端脚本和客户端脚本，而且还可以混合多种脚本语言代码。

3.2　VBScript 的基本元素

VBScript 是 Visual Basic 的子集，编程方法和 Visual Basic 基本相同。本节将学习 VBScript 中的数据类型、运算符、常量、变量和数组等基本元素的概念和使用方法。

3.2.1　数据类型

VBScript 只有一种数据类型，即 Variant 类型，也叫做变体类型。Variant 类型可以在不同的场合代表不同类型的数据。例如，Variant 类型用于数字时，将作为数值处理；用于字符串时，将作为字符串处理。

注意：

由于 Variant 类型是 VBScript 中唯一的数据类型，因此也是 VBScript 中所有函数返回值的数据类型。

大多数情况下，Variant 类型会按照最适用于其包含的数据方式进行操作。例如：

```
Variable=2004               'VBScript 会把 Variable 当成整数对待
Variable= "2004"            'VBScript 会把 Variable 当成字符串对待
Variable=#2004-07-13#       'VBScript 会把 Variable 当成日期对待
```

VBScript 还会根据代码的上下文自动转换数据的子类型。例如 Variable_1="2004"，这里把 Variable_1 看作为一个字符串，而如果下面有一句 Variable_2= Variable_1+3，这时 VBScript 就会自动将 Variable_1 转换为整数变量，然后参与运算。

根据 Variant 类型所能够包含的数值信息类型的不同，可以将这种特殊的数据类型细分为多种子类型，各子类型名称及其说明如表 3-1 所示。

表 3-1　Variant 类型的子类型

子　类　型	说　　明
Empty	声明一个变量后，如果还没有初始化，则该变量的值是 Empty。可以用 IsEmpty()函数来测试变量是否已初始化。当变量为 Empty 值时，可以在表达式中使用，至于是将其作为 0 还是作为零长度的字符串来处理，要根据具体的表达式来定。只要将任何值(包括 0、零长度字符串或 Null)赋予变量，Empty 值就会消失；而将关键字 Empty 赋予变量，就可以将变量恢复为 Empty 值
Null	空值，表示不包含任何有效数据。Null 常用于数据库应用程序，表示未知数据或丢失的数据。如果表达式中包含 Null，那么计算结果总是 Null。将 Null、值为 Null 的变量或计算结果为 Null 的表达式作为参数传递给大多数函数，将使函数返回 Null。可以使用 IsNull()函数来测试表达式是否包含任何有效的数据
Boolean	包含逻辑值，只有 True 或 False 这两个值
Byte	表示 0~255 之间的整数
Integer	表示－32 768~32 767 之间的整数
Currency	表示－922 337 203 685 477.580 8~922 337 203 685 477.580 7 之间的数。Currency 是一个精确的定点类型，适用于货币运算
Long	表示－2 147 483 648~2 147 483 647 之间的整数
Single	单精度浮点数，负数范围从－3.402 823E38~－1.401 298E－45，正数范围为 1.401 298E－45~3.402 823E38
Double	双精度浮点数，负数范围为－1.797 693 134 862 32E308~－4.940 656 454 841 247E－324，正数范围为 4.940 656 458 412 47E－324~1.797 693 134 862 32E308
Date(Time)	表示日期数值，日期范围从公元 100 年 1 月 1 日至公元 9999 年 12 月 31 日。时间值从 00:00:00 至 23:59:59。在代码中使用日期和时间值时，必须用一对#号将其括起来，如#3－10－2007 4:20:16 PM#等。AM 表示上午，PM 表示下午
String	表示字符串数值，字符串的最大长度可为 20 亿个字符
Object	引用程序所能识别的任何对象
Error	包含错误号

注意：

在使用 Variant 类型的数据子类型时，可以使用转换函数来转换数据的子类型，也可以

使用 VarType 函数返回数据的 Variant 子类型。关于这些函数的详细信息，可查看第 3.5 节的相关内容。

3.2.2　运算符

VBScript 提供了 4 种类型的运算符，即算术运算符、连接运算符、关系运算符和逻辑运算符。将运算符和操作数连接起来，就构成了表达式。

1. 算术运算符

算术运算符就是日常的数学运算符，如加(+)、减(－)、乘(*)、除(/)、整除(\)、取模(Mod)、指数(^)和一元减(数值取反)。

2. 连接运算符

连接运算是将两个字符表达式连接起来，生成一个新的字符串。连接运算符有+和&两个。

使用&运算符时，参与连接的两个表达式可以不全是字符串，即&运算符能强制性地将两个表达式的值作为字符串连接。例如：

```
<%
money=56
strTemp="应收金额="&money
%>
```

而使用+运算符时，操作数必须是字符串。例如：

```
<%
money="56"
strTemp="应收金额="+money
%>
```

3. 关系运算符

关系运算符用来比较两个表达式的值的大小，如大于(>)、小于(<)、大于等于(>=)、小于等于(<=)、不等于(< >)和等于(=)。关系运算的结果是逻辑值 True 或 False。关系运算可用于数值间的比较，也可用于字符串间的比较。当用于字符串间的比较时，将按 ASCII 码值的大小由左向右依次逐个比较字符，直到比较出结果为止。

4. 逻辑运算符

逻辑运算通常也称为布尔运算，专门用于逻辑值之间的运算。逻辑运算的各运算符及其含义如表 3-2 所示。

表 3-2　逻辑运算符

运 算 符	含　　义	举　　例	结　　果	说　　明	优先级
Not	逻辑非	Not(3>1)	False	3 大于 1 为真，取反后为假	高
And	逻辑与	(3>1) And(2<4)	True	两个表达式的值都为真时才为真	↑
Or	逻辑或	(3>1) Or(2=4)	True	两个表达式的值有一个为真即为真	│
Xor	逻辑异或	(3>1) Xor(2<4)	False	两个表达式的值有一个且只有一个为真时才为真	│
Eqv	逻辑等于	(3-1) Eqv(4-2)	True	仅当两个表达式的值相同时才为真	│
Imp	逻辑蕴涵	(3>1) Imp(2<4)	False	仅当第一个表达式的值为真时才为真	低

5. 运算符的优先级

当一个表达式包含有多个运算符时，执行运算的符号有一个固定的优先计算顺序，如下所示：

算术运算符>连接运算符>关系运算符>逻辑运算符

对于同优先级的运算符，以从左到右的顺序进行计算。在表达式中，可以使用括号改变计算的优先顺序，强令表达式的某些部分优先运算。括号内的运算总是优先于括号外的运算。

注意：

算术运算符之间的优先顺序是：指数、一元减、乘除、取模和加减。逻辑运算符之间的优先顺序按表 3-2 所示从上到下逐渐降低。

3.2.3　常量

常量就是拥有固定数值的名称，常量可以代表字符串、数字等常数。常量一经声明，在程序执行期间，其值不会发生改变。

声明常量后可以在程序的任何部分使用该常量来代表特定的数值，从而方便程序的编写。例如，在计算程序中常用 PI 来表示 π 的近似值 3.141 592 6，这样即不容易出错，程序也更加简洁明了，并且在程序的其他地方就可以使用 PI 来表示 π 的近似值了。例如：

```
<%
Const PI=3.1415926                    '指定 PI 为常量，其值为 3.141 592 6
```

```
S=PI*R^2                         '求半径为 R 的圆的面积，并将值赋给 S
%>
```

注意：

如果要在多个 ASP 文件中使用一些相同的常量，则可以把常量定义放在单独的文件中，然后在所有使用这些常量的 ASP 文件中包含这些定义即可。

3.2.4　变量

变量用于引用计算机内存地址，该地址可以存储程序运行时可更改的程序信息。在程序中，常用变量来临时存储数据。

1. 声明变量

声明变量有显式声明和隐式声明两种方式。显式声明要用到 Dim 语句、Public 语句和 Private 语句。例如：

```
<% Dim strUserName %>
```

声明多个变量时，可使用逗号来分隔变量。例如：

```
<% Dim strUserName,strPassword,b %>
```

隐式声明变量没有声明变量语句，而直接使用变量名。VBScript 在使用变量时，就会自动创建该变量。例如：

```
<% dtmToay =Now() %>
当前日期和时间是<% =dtmToday %>
```

2. 强制显式声明

变量采用隐式声明的方法很方便，但常常会导致难以查找的错误。例如：

```
<% dtmToday=Now() %>
当前日期和时间是<% =dtmToday %>
```

以上代码看起来好像没有问题，结果应该和前面的代码一样。但由于在 ASP 输出命令中将 dtmToday 写错了，因此无法显示出当前的日期和时间。这是因为当 VBScript 遇到新的名字时，无法确定到底是隐式声明了一个新变量，还是仅仅把现有变量名写错了，于是只好用新名字再创建一个新变量。

为了避免隐式声明时写错变量名引起的问题，VBScript 提供了 Option Explicit 语句来强制显式声明。如果在程序使用该语句，则所有变量必须先声明，然后才能使用，否则会出错。Option Explicit 语句必须位于 ASP 处理命令之后，任何 HTML 文本或脚本命令之前。例如：

```
<% @   Language=VBScript % >
<% Option Explicit %>
<% Dim strUserName %>
```

3. 变量命名约定

给变量起名称时，一定要遵循 VBScript 的标准命名规则，变量命名必须遵循以下几个方面：

- 第一个字符必须是字母。
- 不能包含句点。
- 长度不能超过 255 个字符。
- 在声明的作用域内必须唯一。
- 名字不能和关键字同名。

VBScript 不区分变量名称的大小写。例如，将一个变量命令为 strUserName 和将其命名为 STRUSERNAME 效果是一样的。另外，给变量命名时，要含义清楚，便于记忆。建议尽量按表 3-3 所示的前缀来命名变量，以便通过变量的名称便可获知该变量的子类型。

表 3-3　用来表示子类型的名字前缀

子　类　型	前　　缀	示　　例	子　类　型	前　　缀	示　　例
Integer	int	intAge	Long	lng	lngSum
Currency	cur	curRevenue	Single	sng	sngTotal
Double	dbl	dblTolerance	Byte	byt	bytRasterData
Boolean	bln	blnMarried	String	str	strName
Date(Time)	dtm	dtmStart	Object	obj	objCurrent

4. 变量的作用域与存活期

变量的作用域由声明的位置决定。如果在过程中声明变量，则只有该过程中的代码可访问或更改变量值，此时变量具有局部作用域并称为过程级变量。如果在过程之外声明变量，则该变量可以被脚本中所有过程所识别，称为脚本级变量，具有脚本级作用域。

变量存在的时间称为存活期。脚本级变量的存活期从被声明的那一刻起，直到脚本运行结束。对于过程级变量，其存活期仅是该过程运行的时间，该过程结束后，变量随之消失。在执行过程时，局部变量是理想的临时存储空间。可以在不同过程中使用同名的局部变量，这是因为每个局部变量只被声明它的过程识别。

3.2.5　数组

数组变量是一类具有相同名字，但有不同的下标值的变量，简称数组。数组中的每个元素都用唯一的下标来识别。例如 intAge(9)中，intAge 是数组名，9 是下标。

数组变量的声明同简单变量相同，唯一的区别是声明数组变量时变量名后面带有括号，如下面的代码声明了一个包含 20 个元素的一维数组：

```
Dim lngSum (19)
```

数组的下标是从 0 开始的，数组元素的数目是括号中的显示的数目加 1，所以上面这个数组变量实际上包含 20 个元素。下面的代码可以对上面定义的数组进行赋值：

```
lngSum (0)=1
lngSum (1)=2
lngSum (2)=3
…
lngSum (19)=20
```

具有两个或多个下标的数组称为二维数组或多维数组。声明多维数组时用逗号分隔括号中每个下标，如下面的代码就声明了一个 5 行 8 列的二维数组：

```
Dim intCounters(4,7)
```

如果事先不知道数组的大小，可声明一个动态数组。动态数组可以在运行脚本时根据实际的需要，使用 ReDim 来调整数组的维数和每一维的大小。例如：

```
Dim curRevenue ()
ReDim curRevenue (25)
…
ReDim curRevenue (9,11)
```

注意：

重新调整动态数组大小的次数是没有任何限制的，但是，存储在数组中的当前值都会全部丢失，VBScript 重新将数组元素的值全部置为空。如果希望改变数组大小的同时而又不丢失数组中的数据，则要在 ReDim 语句中带上 Preserve 关键字，如<% ReDim Preserve curRevenue (9,11) %>。但仍要注意的是，将数组的大小调小时，仍将会丢失被删除元素的数据。

3.3　VBScript 常用语句

默认情况下，脚本中的代码总是按书写的先后顺序来执行的。但在实际应用中，通常要根据条件的成立与否来改变代码的执行顺序，这时就要使用控制结构。

在 VBScript 中，控制结构有判定结构和循环结构两种。判定结构主要是根据判断的结果来决定执行哪部分代码，它又分为条件结构和选择结构两种。

3.3.1　条件语句

条件语句用于判断条件是 True 还是 False，并且根据判断结果来指定要运行的语句(语

句既可为单条语句，也可以是由多条语句组成的复合语句)。条件语句有以下几种形式。

形式 1：

```
If 条件表达式 Then 语句 End If
```

如果条件表达式的值为 True，则执行 Then 后的语句，否则跳出该条件语句。

形式 2：

```
If 条件表达式 Then 语句 1　Else 语句 2　End If
```

如果条件表达式的值为 True，则执行 Then 后的语句 1，否则转去执行 Else 后的语句 2。

形式 3：

```
If 条件表达式 1Then
语句 1
ElseIf 条件表达式 2 Then
语句 2
…
Else
语句 n+1
End If
```

如果条件表达式 1 的值为 True，则执行语句 1，然后跳出 If 语句；如果条件表达式 2 的值为 True，则执行语句 2，然后跳出 If 语句……若所有条件表达式的值都不为 True，执行语句 n+1。

【练习 3-2】利用条件语句，创建一个成绩判断程序，运行效果如图 3-3 所示。

(1) 创建一个名为 program.html 的网页，其代码如下所示:

```
<Html><Body>
    <Script Language="VBScript" >                    '指定下面语句所采用的脚本语言为 VBScript
    Dim intGrade                                     '定义一个存放分数的变量 intGrade
    intGrade = InputBox("请输入分数")
    '弹出请求输入分数的对话框，并将输入分数赋值给 intGrade
    If intGrade>100 Or intGrade<0 Then
        Msgbox("输入成绩有误")
    '对输入的分数进行判断，如果大于 100 或小于 0，则弹出信息框提示“输入成绩有误”
    ElseIf intGrade>=85 Then
        Msgbox("成绩优秀，还要继续努力加油！")
        '如分数大于 85，则弹出信息框提示“成绩优秀，还要继续努力加油！”
    ElseIf intGrade>=60 Then
        Msgbox("成绩及格")
        '如分数大于 60，则弹出信息框提示“成绩及格”
    Else
        Msgbox("不及格")
        '否则，弹出信息框提示“不及格”
```

```
        End If
        </Script>
</Body></Html>
```

(2)运行 program.html 网页文档，效果如图 3-3 所示。



图 3-3　program.html 的运行效果

3.3.2　选择语句

选择语句 Select Case 是 If 语句多条件时的另外一种表示方法，可在多个执行语句中有选择地执行其中的一个。它的优点是程序更简洁易读，语法结构如下：

```
Select Case  表达式
Case  结果 1
 语句 1
Case  结果 2
 语句 2
…
Case  结果 n
 语句 n
Case Else
 语句 n+1
```

```
End Select
```

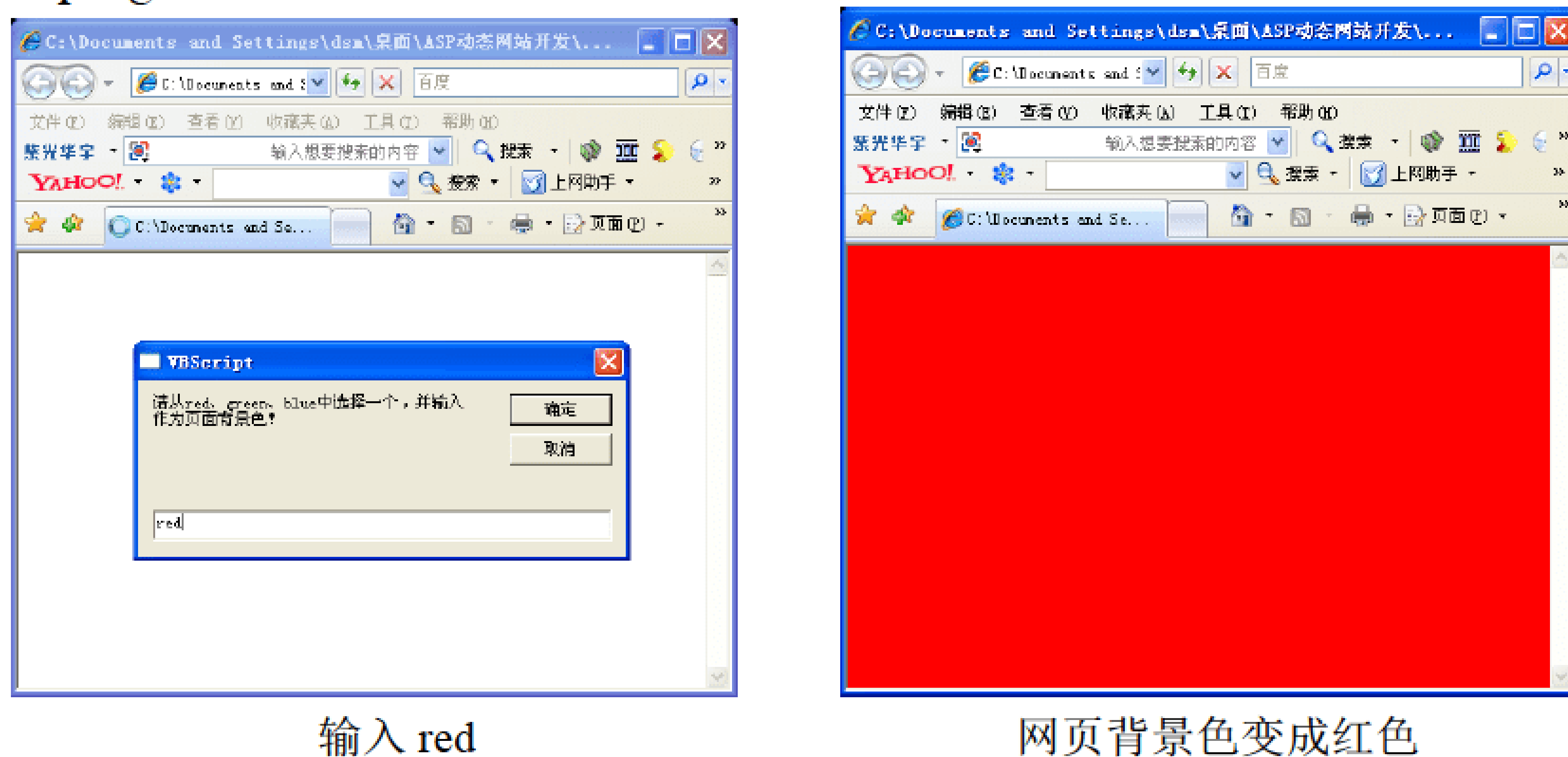
VBScript 首先对表达式进行运算，这个运算可以为数值运算或字符串运算，然后将运算结果依次与结果 1 到结果 n 作比较，当找到与计算结果相等的结果时就执行该语句，执行完毕后就跳出 Select Case 语句。而当运算结果与所有的结果都不相等时，就执行 Case Else 后面的执行语句 n+1。

【练习 3-3】利用选择语句，创建一个可以根据用户的选择设置页面背景颜色的程序，运行效果如图 3-4 所示。

(1) 创建一个名为 program2.html 的网页，其代码如下所示：

```
<Html><Body>
    <Script Language="VBScript" >                '指定下面语句所采用的脚本语言为 VBScript
    Dim strColor                                 '定义一个存放颜色的变量 strColor
    strColor=inputbox("请从 red、green、blue 中选择一个，并输入作为页面背景色！")
    Select Case strColor
        Case "red"     document.bgColor = "red"   '如果 strColor 值为 red，设置页面背景为红色
        Case "green"   document.bgColor = "green" '如果 strColor 值为 green，设置页面背景为绿色
        Case "blue"    document.bgColor = "blue"  '如果 strColor 值为 blue，设置页面背景为蓝色
        Case Else      MsgBox "输入有误"   '如果不是上面 3 种颜色，则提示“输入有误”
    End Select
    </Script>
</Body></Html>
```

(2)运行 program2.html 网页文档，效果如图 3-4 所示。



输入 red　　　　网页背景色变成红色

图 3-4　program2.html 的运行效果

3.3.3　循环语句

循环结构允许重复执行一行或数行代码。在 VBScript 中，提供了 3 种不同风格的循环语句，即 Do 循环、For 循环和 For Each 循环。

1. Do 循环

Do…Loop 循环是一种条件型的循环，当条件为 True 时或条件变为 True 之前，重复执行语句块，该循环共有 3 种形式。

形式 1：

```
Do While 条件表达式
语句
Loop
```

VBScript 首先检查条件表达式的值是否为 True，如果为 True 才会进入循环执行语句。另外，也可以对语句顺序进行调整，使它先进入循环执行一次后再对条件进行判断。

形式 2：

```
Do
语句
Loop While 条件表达式
```

如果把 Do 循环中的 While 换为 Until，则程序的运行过程和前面类似。不同的是，只要条件为 False 就执行循环。

形式 3：

```
Do
语句
Loop Until 条件表达式
```

注意：

Do 循环语句中可以使用 Exit Do 语句强行中止循环。

【练习 3-4】利用 Do 循环语句，创建一个可以显示循环语句的执行过程的程序，运行效果如图 3-5 所示。

(1) 创建一个名为 program3.asp 的网页，其代码如下(如图 3-6 所示):

```
<%
intNum=1
Do While intNum<7
Response.Write "循环语句正在执行第"&intNum&"次循环"
Response.Write "<br>"                    '在屏幕上输出一个回车
intNum=intNum+1
Loop
%>
```

(2) 将 program3.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/program3.asp，按 Enter 键，效果如图 3-5 所示。

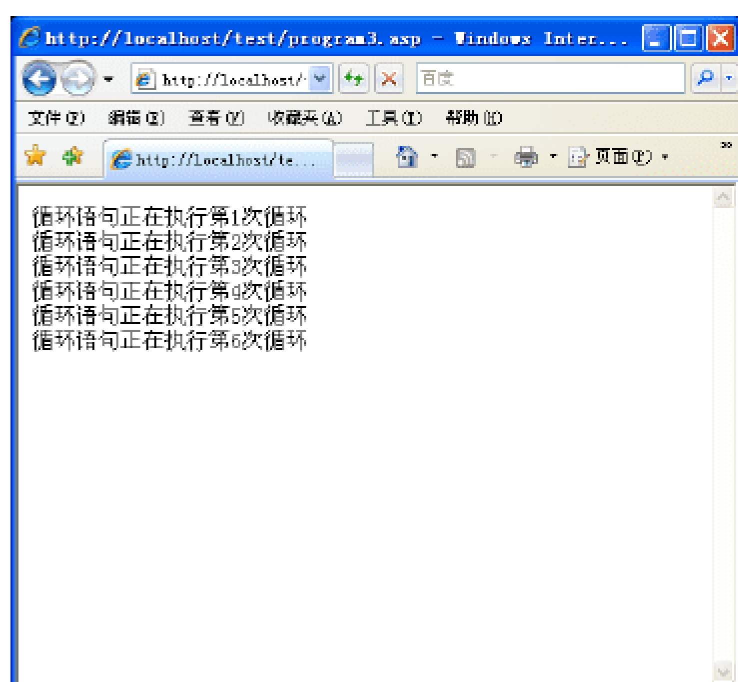


图 3-5　program3.asp 的运行效果

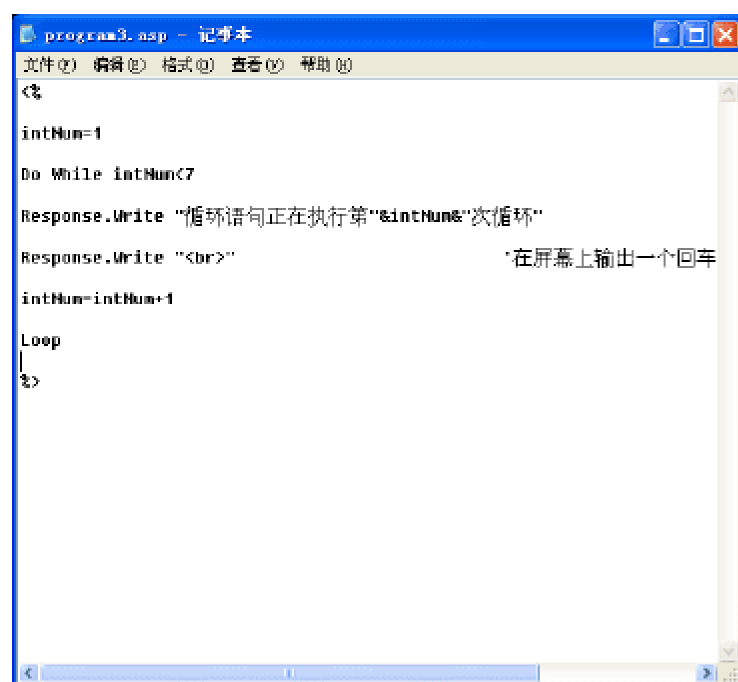


图 3-6　program3.asp 的代码

2. For 循环

For…Next 循环是一种强制性的循环，用于将循环体运行指定的次数。在 For 循环中有一个计数器变量，每重复一次循环，该变量的值都会增加或减少，其语法结构如下：

```
For 循环变量=初始值 To 结束值 [Step 步长值]
 执行语句
Next
```

For…Next 循环语句的执行步骤如下：

(1) 将循环变量设为初始值。

(2) 测试循环变量是否大于结束值，若是，则退出循环(若步长值为负，则测试循环变量是否小于结束值)，否则执行循环中的语句。

(3) 执行语句运行完毕到 Next 语句时，VBScript 将循环变量值与步长值相加。

(4) 从 Next 语句跳转到 For 语句继续执行。

注意：

如果省略 Step 子句，那么步长的默认值是 1。

【练习 3-5】利用 For 循环语句，创建一个可以求出并显示 1~100 的奇数和的网页程序，运行效果如图 3-7 所示。

(1) 创建一个名为 program4.asp 的网页，其代码如下(如图 3-8 所示):

```
<%
Dim intNum,i,intSum
intNum=1
intSum=0
For I=1 To 100 Step 2
Response.Write "循环语句正在执行第"&intNum&"遍循环"&"<br>"
Response.Write "第"&intNum&"个欲相加的 I 值为"&i&"<br>"
Response.Write "SUM="&intSum&"+"&i&"="&intSum+i&"<br>"
intSum=intSum+i
```

```
intNum=intNum+1
Next
%>
```

(2) 将 program4.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/program4.asp 并按 Enter 键，效果如图 3-6 所示。

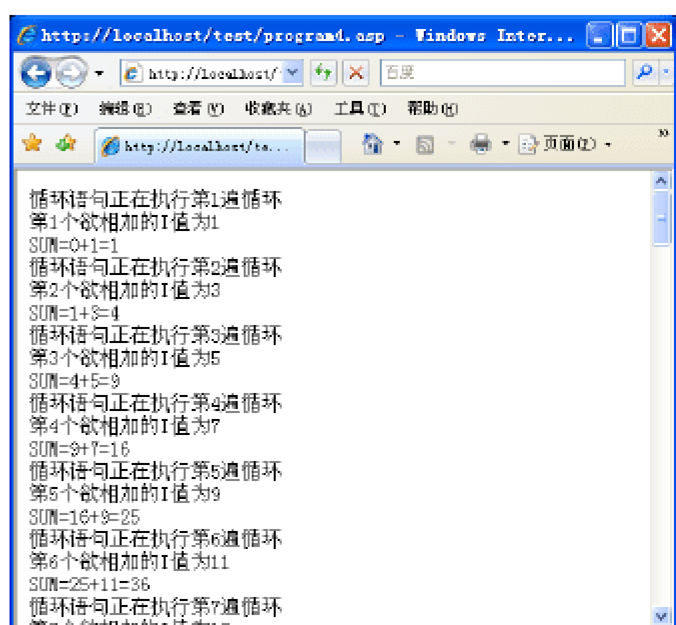


图 3-7　program4.html 的运行效果

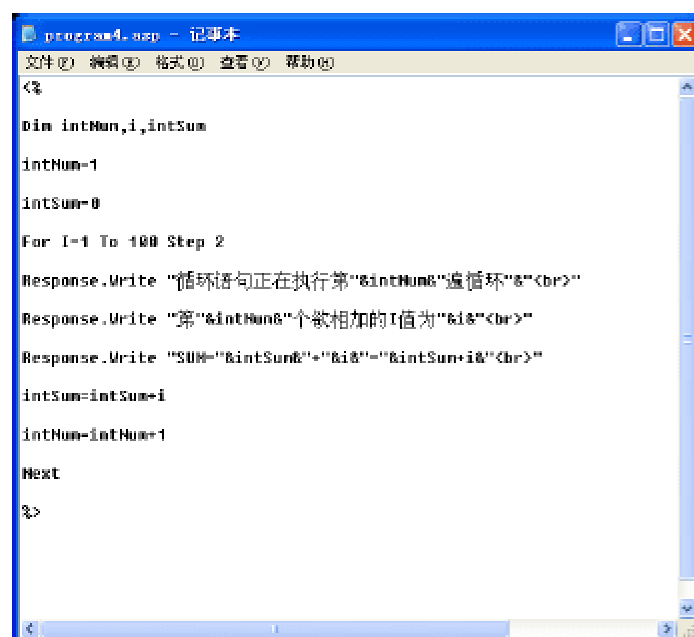


图 3-8　program4.html 的代码

3. For Each 循环

For Each…In 循环只针对数组或对象集合中每一个元素的遍历循环。执行循环的次数不再需要指定，只要有一个元素就循环一次，其语法结构如下：

```
For Each 元素 In 集合
语句
Next
```

如果不知道集合中有多少个元素，则用 For Each 循环非常方便。如下例可列举使用 HMTL 表单提交的所有数值：

```
<%
For Each item In Request.Form
 Response.Write Request.Form(item)
Next
%>
```

注意：

For 循环语句中可以使用 Exit For 语句强行中止循环。

3.4　过程与函数

过程是用来执行特定任务的独立的程序代码。使用过程，可以将程序划分成一个个较

小的逻辑单元，过程中的代码能够被反复调用，这样可以减少不必要的重复。可以将过程定义放在调用该过程的同一个 ASP 文件中，也可以将常用过程放在共享的文件中，并使用 #include 命令将该文件包含在调用过程的 ASP 文件中。

VBScript 根据是否返回值将过程划分为 Sub 过程(子过程)和 Function 过程(函数)两种。Sub 过程只执行程序而不返回值，因而不能用于表达式中，而 Function 函数可以将执行代码后的结果返回给请求程序。

3.4.1　Sub 过程

Sub 过程是一种可以获取参数，执行一系列语句以及可改变其参数值的独立过程。Sub 过程可以使用参数(由调用过程传递的常数、变量或表达式)，参数在调用过程和被调用过程之间传递信息。如果 Sub 过程无任何参数，则 Sub 语句必须包含空括号()。其语法结构如下：

```
Sub 子程序名(参数 1,参数 2,…)
…
End Sub
```

定义一个 Sub 过程后，就可以在程序代码中调用它。Sub 过程的调用有两种方式。一种是使用 Call 语句，它要求将所有参数包含在括号之中，其语法结构如下：

```
Call 子程序名 (参数 1,参数 2,…)
```

另一种是直接使用子过程名，只需输入过程名及所有参数值，参数值之间使用逗号分隔。其语法结构如下：

```
子过程名 参数 1,参数 2,…
```

注意：

使用 Exit Sub 语句可以立即从 Sub 过程中退出，程序继续执行调用 Sub 过程之后的语句，在 Sub 过程的任意位置可以出现任意个 Exit Sub 语句。

【练习 3-6】建立一个可将参数在弹出式信息对话框中显示的 Sub 过程，并在程序中多次调用它，程序运行效果如图 3-9 所示。

(1) 创建一个名为 program5.html 的网页，其代码如下所示：

```
<Script Language="VBScript" >
Call output("第一次调用过程")
output  "再试一次调用过程"

Sub output(strText)
   MsgBox strText
End Sub
</Script>
```

(2) 运行 program5.html 网页文档，效果如图 3-9 所示。

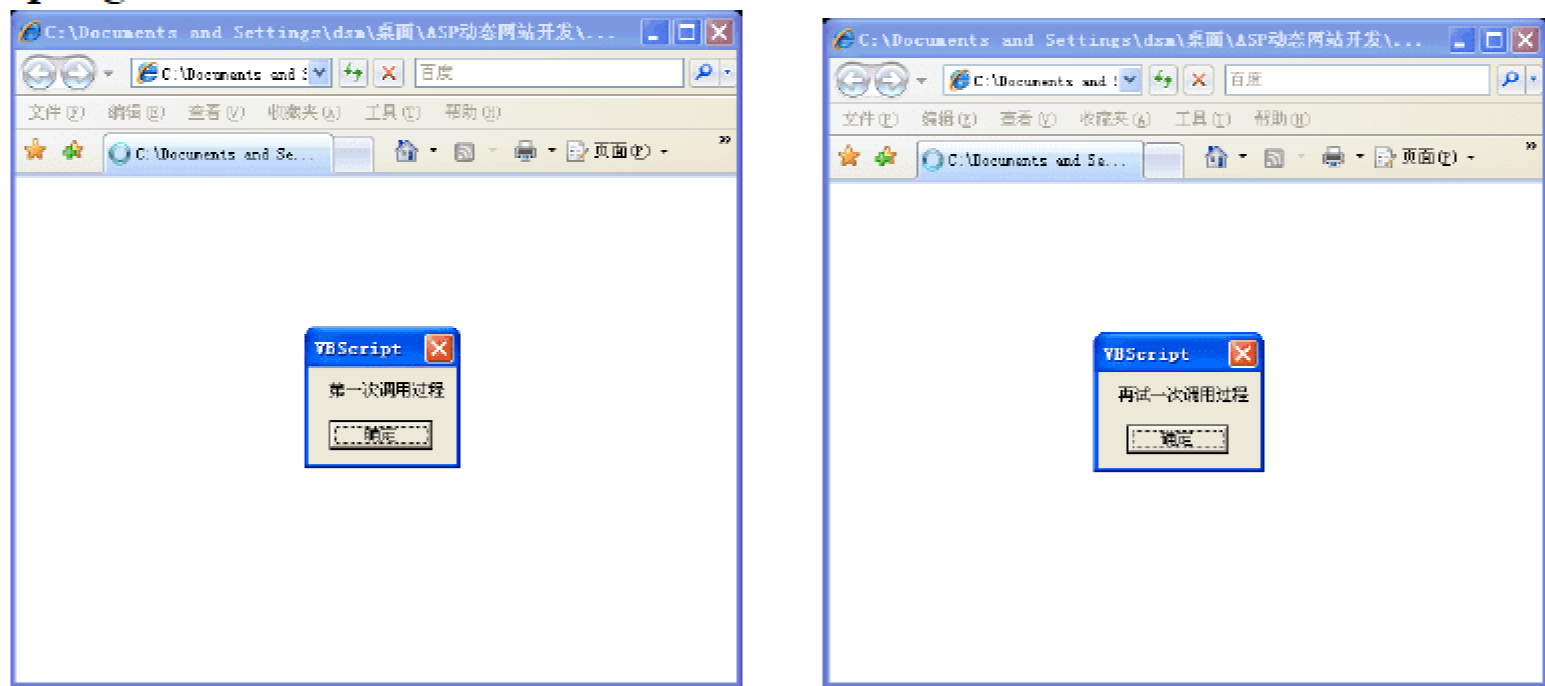


图 3-9　program5.html 的运行效果

注意：

在 Sub 过程中使用的变量分为两类，一类是在过程内显式声明的，另一类则不是。在过程内显式声明的变量(使用 Dim 或等效方法)是局部变量，对于那些没有在过程中显式声明的变量也是局部变量，除非在该过程外更高级别的位置显式地声明它们。

3.4.2　Function 函数

函数和子过程一样，也是用来完成特定功能的独立的程序代码，可以读取参数、执行一系列语句并改变参数的值。但函数有一个最重要的特点，就是调用时将返回一个值。函数的语法结构如下：

```
Function 函数名(参数 1,参数 2,…)
[语句]
…
函数名=表达式
[语句]
End Function
```

与 Sub 过程类似，其中“参数 1,参数 2,…”是调用时传递的常数、变量或表达式，如果无任何参数，则 Function 语句必须使用空括号。与 Sub 过程不同的是，Function 函数通过函数名返回一个值，这个值是在函数体中赋给函数名的，Function 返回值的数据类型是 Variant。

“函数名=表达式”用于为函数设置返回值，该值将返回给调用的语句，函数中至少要含有一条这样的语句。

Function 函数只有通过直接引用函数名实现函数的调用，而且函数名必须用在变量赋值语句的右端或表达式中。调用函数时，参数要放在一对括号中，这样就可以将它们和表达式的其他部分分开。例如：

```
temp = Celsius(60)
```

注意：

使用 Exit Function 语句可以从 Function 过程中立即退出，程序继续执行调用 Function 过程之后的语句，但执行前必须为函数赋值，否则就会出错。在 Function 过程的任意位置可以出现任意个 Exit Function 语句。

【练习 3-7】创建一个函数，将华氏温度换算为摄氏温度，程序运行效果如图 3-10 所示。

(1) 创建一个名为 program6.html 的网页，其代码如下所示：

```
<Script Language="VBScript" >
temp = InputBox("请输入华氏温度。", 1)
MsgBox "温度为 " & Celsius(temp) & " 摄氏度。"
Function Celsius(fDegrees)
      Celsius = (fDegrees - 32) * 5 / 9
End Function
</Script>
```

(2) 运行 program6.html 网页，效果如图 3-10 所示。

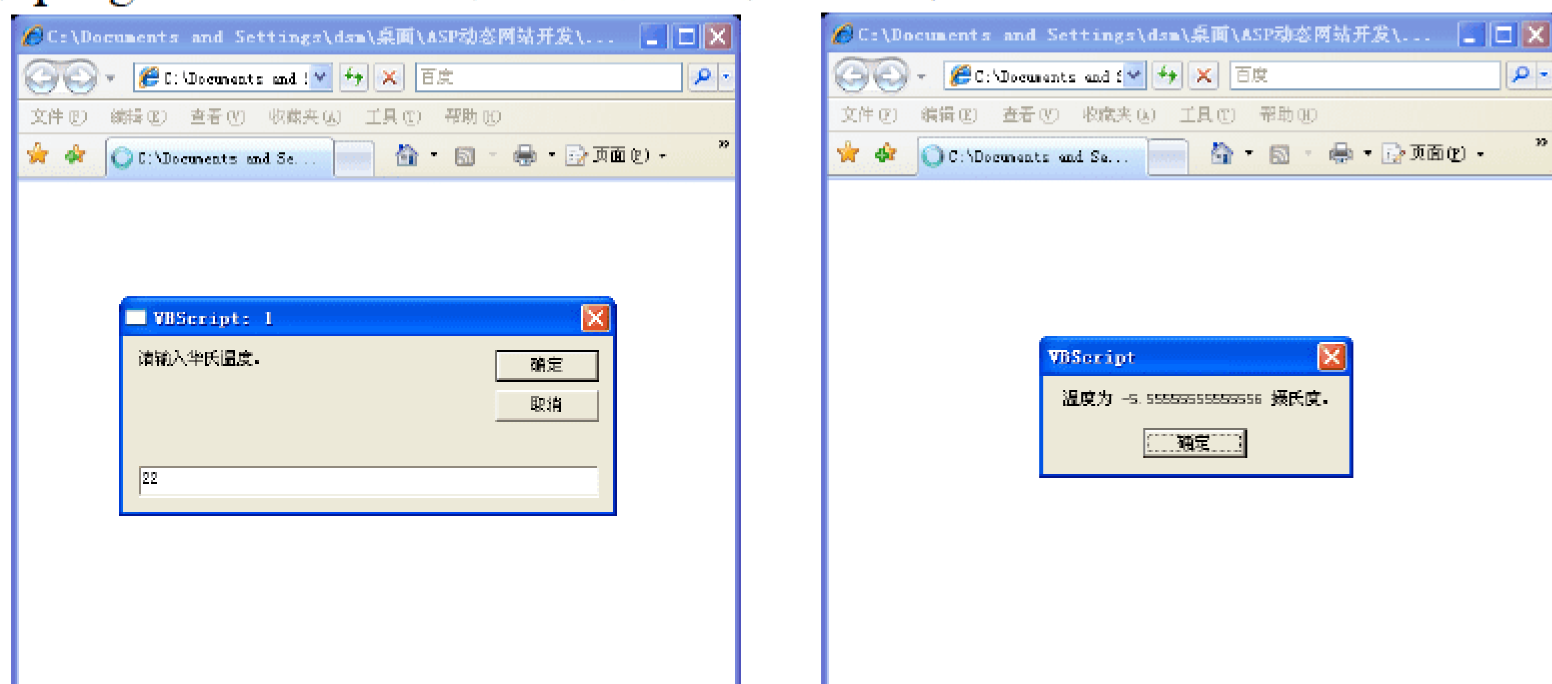


图 3-10　program6.html 的运行效果

3.4.3　按地址传递和按值传递

默认情况下，参数按地址传递，即过程按变量的内存地址去访问实际变量的内容。这样将变量传递给过程时，通过过程就可以改变变量的值。地址传递的关键字为 ByRef。

另外，传递参数还可以按数值来传递。按值传递参数时，传递的只是变量的副本。如果过程中改变了这个值，则所作的变动只影响副本而不会影响到变量本身。按值传递的关键字是 ByVal。

【练习 3-8】建立两个分别按地址传递和按值传递的过程，比较其异同，程序运行的结果如图 3-11 所示。

(1) 创建一个名为 program7.asp 的网页，其代码如下所示(如图 3-12 所示):

```
<%
Sub TestByVal(ByVal X)
X=X*33
End Sub
Sub TestByRef(ByRef X)
X=X/10
End Sub

intNum=60
Response.Write "原测试数值为"&intNum&"<br>"&"<br>"
Call TestByVal(intNum)
Response.Write "使用 ByVal 关键字，参数按值传递调用过程后，_测试变量的值为
  "&intNum&"<br>"&"<br>"
Call TestByRef(intNum)
Response.Write "使用 ByRef 关键字，参数按地址传递调用过程后，_测试变量的值为"&intNum
%>
```

(2) 将 program7.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/program7.asp 并按 Enter 键，效果如图 3-12 所示。

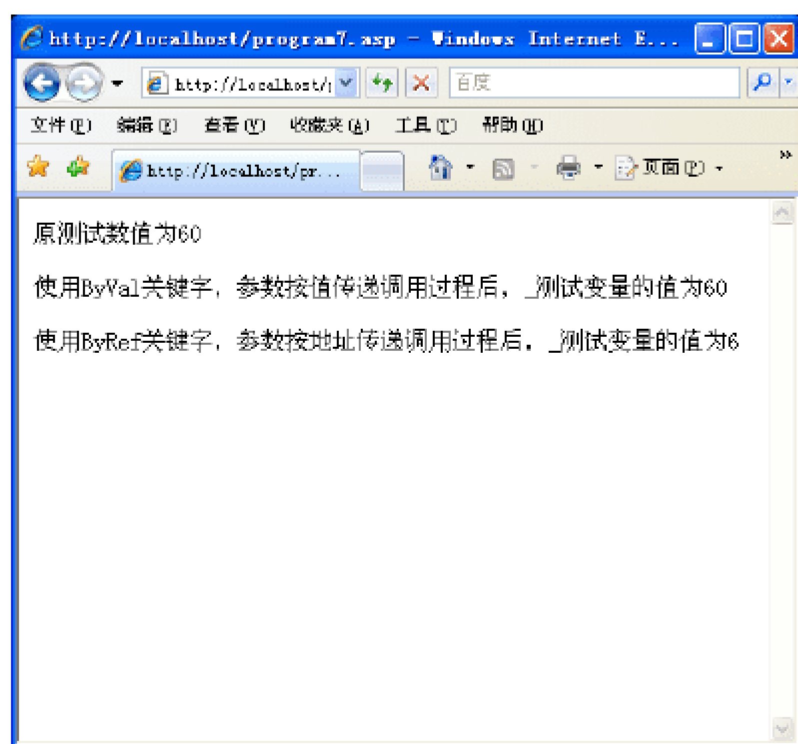


图 3-11　program7.asp 的运行效果



图 3-12　program7.asp 的代码

从上例的运行效果可看出，按值传递后，变量 intVar 的值保持不变，仍为 60，而按地址传递后，值将变为 6。

3.4.4　数组参数

数组作为参数传递时，与普通参数有些不同。在过程中声明数组参数时，必须略去数组的长度，即设置一个动态数组。在调用语句中，只需给出要传递的数组名即可，不必包

括数组的下标及圆括号。

【练习 3-9】用产生随机数的方法给一个数组赋值，并调用过程来将其中的元素按从小到大的顺序排序，程序运行效果如图 3-13 所示。

(1) 创建一个名为 program8.asp 的网页，其代码如下所示(如图 3-14 所示):

```
<%
Sub Sort (arr(),arrnum)
 Dim I,J,intTemp
 For I=0 to arrnum
       For J=0 to arrnum
             If arr(I)>arr(J) Then
             intTemp=arr(I)
             arr(I)=arr(J)
             arr(J)=intTemp
             End If
       Next
 Next
End Sub
Dim arrTest(9),I
For I=0 to 9
 arrTest(I)=Int(Rnd()*100+1)
Next
Sort arrTest,9
For I=0 To 9
 Response.Write arrTest(I)&"<br>"
Next
%>
```

(2) 将 program8.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/program8.asp 并按 Enter 键，效果如图 3-13 所示。

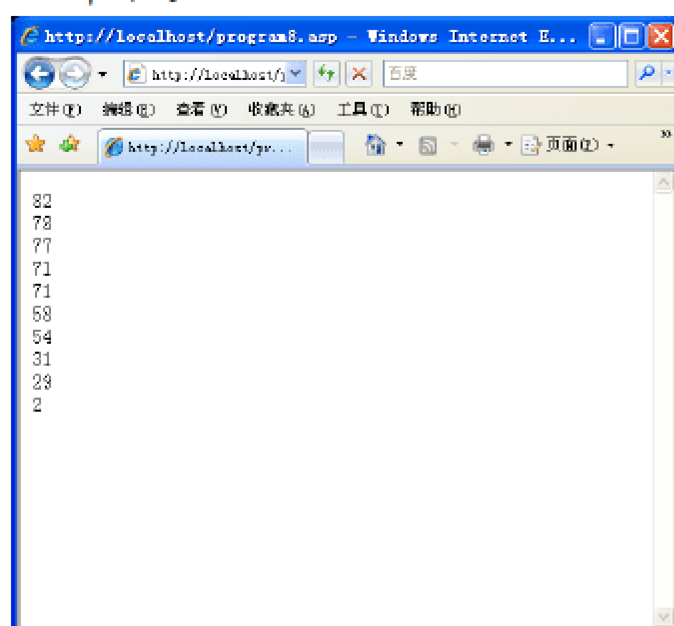


图 3-13　程序运行效果

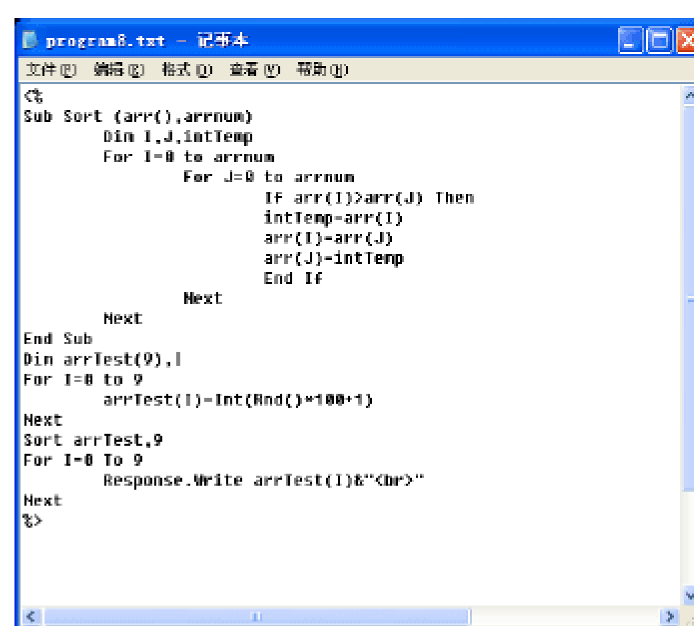


图 3-14　输入代码

3.5　VBScript 函数

VBScript 把一些最常使用的功能整理起来，编制好了相应的处理程序，将它们以函数的形式提供使用。恰当的使用函数可以节省大量的时间，如表 3-4 所示对 VBScript 提供的函数做了概括。

表 3-4　VBScript 函数概括

函　　数	说　　明
Abs	返回一个数的绝对值，如 Abs(－1)和 Abs(1)都返回 1
Array	返回一个 Variant 值，其中包含一个数组
Asc	返回与字符串的第一个字母相关的 ANSI 字符编码，如 Asc("Body")返回 66
Atn	返回一个数的反正切值。如 4*Atn(1)得到 π 的值 3.141 592 653 589 79
Cbool	计算表达式的布尔值。如果表达式的值为 0，则返回 False，否则返回 True，如 CBool(9)返回 True
Cbyte	将表达式转换为 Byte 子类型，如 CByte(45.678 9)返回 46
Ccur	将表达式转换为 Currency 子类型，如 CCur(123.456 789)返回 123.456 8
Cdate	将表达式转换为 Date 子类型，如 Cdate("July 21,2004")返回 2004-7-21
CDbl	将表达式转换为 Double 子类型
Chr	返回与指定的 ASCII 字符代码相对应的字符，如 Chr(66)返回 B
Cint	将表达式转换为 Integer 子类型，如 CInt(456.78)返回 457
CLng	将表达式转换为 Long 子类型，如 CLng(123 456.78)返回 123 457
Cos	返回某个角的余弦值
CreateObject	创建并返回对象实例。注意，在 ASP 中不要用该函数来创对象实例
CSng	将表达式转换为 Single 子类型
CStr	将表达式转换为 String 子类型，如 str=CStr(87.65)，则 str 值为字符串 87.65
Date	返回当前系统日期
DateAdd	返回已添加指定时间间隔的日期，如 DateAdd("m",2,"21-July-2004")将 2004 年 7 月 21 日加两个月，得到 2004-9-21
DateDiff	返回两个日期间的时间间隔，如 DateDiff("d","21-7-2004","23-7-2004")将返回 2
DatePart	返回给定日期的指定部分，如 DatePart("yyyy","21-July-2004")返回 2004
DateSerial	使用指定的年、月、日返回 Date 子类型，如 DateSerial(1996-10,8-2,1-1)将返回 1986-5-31
DateValue	返回 Date 子类型，如 DateValue("September 11,1963")将返回 1963-9-11
Day	返回 1~31 之间的一个整数，表示某月中的一天，如 Day("July 21,2004")将返回 21
Eval	计算一个表达式的值并返回结果

(续表)

函　数	说　明
Exp	返回 e(自然对数的底)的幂次方
Filter	返回下标从 0 开始的数组，包含基于特定过滤条件的字符串数组的子集
Int、Fix	返回数字的整数部分，Int 和 Fix 的区别在于：如果参数为负数，则 Int 返回小于或等于参数的第一个负整数，而 Fix 返回大于或等于参数的第一个负整数。例如，Int(－6.4)将返回－7，而 Fix(－6.4)将返回－6
FormatCurrency	返回一个表达式，该表达式被格式化为货币值(使用系统控制面板中定义的货币符号)。例如，当系统控制面板中定义的货币符号是人民币符号时，FormatCurrency(2000)将返回￥2000.00
FormatDateTime	格式化日期和时间。例如，FormatDateTime(Now,0)返回值为 2004-7-15 11:25:45，而 FormatDateTime(Now,1)返回值为 2004 年 7 月 15 日
FormatNumber	格式化一个数值。例如，FormatNumber(1233.4567,2)返回值为带两位小数点的数 1233.46，FormatNumber(－0.1234,3,-1)返回值为－0.123(含小数点前的 0)，而 FormatNumber(－0.1234,3,0)返回值为–.123(不含小数点前的 0)
FormatPercent	格式化为以%符号结尾的百分比，如 FormatPercent(2/32)返回 6.25%
GetObject	访问文件中的自动化对象，并将该对象赋给对象变量。注意，在 ASP 中不要用该函数来创建对象实例
Hex	返回表示十六进制数字值的字符串，如 Hex(459)返回 1CB
Hour	返回 0~23 之间的一个整数，表示一天中的某一小时
InpuBox	显示一个文本框，提示用户输入一个数据
InStr	返回某字符串在另一字符串中第一次出现的位置
InStrRev	返回某字符串在另一个字符串中出现的从结尾算起的位置
IsArray	返回布尔值，确定一个变量是否为数组
IsDate	返回布尔值，确定表达式是否可以转换为日期
IsEmpty	返回布尔值，确定一个变量是否为空
IsNull	返回布尔值，确定一个表达式是否包含无效的数据
IsNumeric	返回布尔值，确定一个表达式是否为数字
IsObject	返回布尔值，确定一个表达式是否引用了有效的对象
Join	将数组中的多个子字符串合成一个字符串
LBound	返回数组某一维的下界
LCase	返回字符串的小写形式
Left	返回指定数目的从字符串的左边算起的字符
Len	返回字符串内字符的数目

(续表)

函　　数	说　　明
LoadPicture	返回图片对象。LoadPicture 识别的图形格式有位图文件(.bmp)、图标文件(.ico)、行程编码文件(.rle)、图元文件(.wmf)、增强型图元文件(.emf)、GIF 文件(.gif)和 JPEG 文件(.jpg)
Log	返回数值的自然对数
Ltrim、Rtrim 和 Trim	截去字符串中的前导空格(Ltrim)、后续空格(Rtrim)或前导与后续空格(Trim)
Mid	从字符串中返回指定数目的字符
Minute	返回 0~59 之间的一个整数，表示一小时内的某一分钟
Month	返回 1~12 之间的一个整数，表示一年中的某月
MonthName	返回代表指定月份的字符串
MsgBox	显示一个信息对话框
Now	根据计算机系统设定的日期和时间返回当前的日期和时间值
Oct	返回表示八进制数值的字符串，如 Oct(8)返回 10
Replace	将字符串内的子字符串替换为指定的串
RGB	返回代表 RGB 颜色值的整数
Right	从字符串右边返回指定数目的字符
Rnd	返回一个随机数
Round	返回按指定位数进行四舍五入的数值，如 Round(3.141 59,2)返回 3.14
ScriptEngine	返回一个代表当前使用的脚本程序语言的字符串
Second	返回 0~59 之间的一个整数，表示一分钟内的某一秒
Sgn	返回表示数字符号的整数，如 Sgn(2)返回 1，Sgn(0)返回 0， Sgn(－3)返回－1
Sin	返回一个角度的正弦值
Space	返回由指定数目的空格组成的字符串
Split	返回基于 0 的一维数组，其中包含指定数目的子字符串
Sqr	返回数值的平方根
StrComp	比较两个字符串，并返回比较结果
String	返回指定长度的、重复字符组成的字符串，如 String(3, "B")返回 BBB
StrReverse	将字符串按反序排列输出
Tan	返回一个角度的正切值
Time	返回目前系统的时间
Timer	返回午夜 12 时以后已经过去的秒数
TimeSerial	返回含指定时、分、秒的时间，如 TimeSerial(12－6,－15,0)返回 5:45:00AM
TimeValue	返回包含时间的 Data 子类型
TypeName	返回一个变量的子类型信息

(续表)

函　　数	说　　明
UBound	返回数组某一维的上界
UCase	返回字符串的大写形式
VarType	返回表示变量子类型的整数
Weekday	返回表示一星期中某天的整数
WeekdayName	返回一个字符串，表示一星期中指定的某一天
Year	返回一个代表某年的整数

3.6　习　　题

3.6.1　填空题

(1) 在 ASP 文件中直接声明主要脚本语言为 VBScript 的语句为__________。

(2) VBScript 只有一种数据类型，即__________类型，也叫做变体类型。

(3) VBScript 包括 4 种类型的运算符，即算术运算符、__________、__________和逻辑运算符。

(4) __________一经声明，在程序执行期间，其值不会发生改变。

(5) VBScript 中声明多个变量时，使用__________分隔变量。

(6) 数组中的每个元素都用唯一的__________来识别。

3.6.2　选择题

(1) 下面关于 VBScript 的标准命名规则的说法不正确的是(　　)。

A. 第一个字符必须是字母或数字

B. 长度不能超过 255 个字符

C. 名字不能和关键字同名

D. 在声明的作用域内必须唯一

(2) 使用(　　)语句可以立即从 Sub 过程中退出，程序继续执行调用 Sub 过程之后的语句。

A. Exit Sub　　B. Exit　　C. </sub>　　D. Loop

(3) 下面不属于变量子类型的有(　　)。

A. Empty 表示数据未被初始化

B. Integer 表示－32，768~32，767 之间的整数

C. Single 表示单精度浮点数，而 Double 表示双精度浮点数

D. String 表示字符串数值，字符串的最大长度可为 200 个字符

3.6.3　问答题

(1) 在编写 ASP 代码时，如何声明所使用的脚本语言种类。

(2) 简述运算符的优先顺序。

(3) 简述什么是变量的作用域。

3.6.4　操作题

(1) 参照本章练习，编写一段 VBScript 程序计算表达式 $11^2+12^2+13^2+\cdots+23^2$ 的值。

(2) 编写一段 VBScript 程序，实现以下功能。弹出对话框请求输入 n 值和 m 值，然后弹出信息框显示(n-1)!+(m+1)!/(n+m)!表达式的计算结果。

第4章　Response和Request对象

在 ASP 中，与客户端的动态交互是通过 Response 和 Request 对象实现的，这两个对象起到了服务器与客户机之间的信息传播作用，其中 Response 对象用于接收客户端浏览器提交的数据，而 Request 对象的功能则是将服务端的数据发送到客户端浏览器。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 了解从浏览器取得数据的方法
- 了解 Request.QueryString 集合读取表单 GET 方法数据
- 了解 Request.Form 集合读取表单 POST 方法数据
- 掌握网页重定位的实现方法
- 掌握利用缓冲区输出数据的方法

4.1　Request 对 象

Request 对象是 ASP 最常用的对象之一，其作用是连接客户端的 Web 页面(HTML 页面)和服务器端的 Web 页面(ASP 页面)，使他们之间可以进行数据交换。Request 对象允许 ASP 查询有关与之交互的客户程序信息，代表由客户程序发出的 HTTP 请求报文。

4.1.1　Request 对象简介

Request 对象可以用于访问任何用 HTTP 请求传递的信息，包括从 HTML 表单的 POST 方法或 GET 方法传递的参数、Cookie 和用户认证。

1. Request 对象的集合

Request 对象提供了 5 个集合，可以用来访问客户端对 Web 服务器请求的各类信息，具体说明如表 4-1 所示。

表 4-1　Request 对象的集合及说明

集 合 名 称	说　　明
ClientCertificate	当客户端访问一个页面或其他资源时，用来向服务器表明身份的客户证书的所有字段或条目的数值集合，每个成员均是只读
Cookies	根据用户的请求，用户系统发出的所有 Cookie 值的集合，这些 Cookie 仅对相应的域有效，每个成员均为只读

(续表)

集 合 名 称	说　　明
Form	METHOD 的属性值为 POST 时，所有作为请求提交的<FORM>段中的 HTML 控件单元的值的集合，每个成员均为只读
QueryString	依附于用户请求的 URL 后面的名称 / 数值或作为请求提交的且 METHOD 属性值为 GET(或者省略其属性)，或<FORM>中所有 HTML 控件单元的值，每个成员均为只读
ServerVariables	随同客户端请求发出的 HTTP 报头值，以及 Web 服务器的几种环境变量的值的集合，每个成员均为只读

2. Request 对象的属性

如表 4-2 所示为 Request 对象唯一的属性及说明，它提供关于用户请求的字节数量的信息，并很少用于 ASP 网页。

表 4-2　Request 对象的属性及说明

属　　性	说　　明
TotalBytes	只读，返回由客户端发出请求的整个字节数量

3. Request 对象的方法

如表 4-3 所示为 Request 对象唯一的方法及说明，它允许访问从一个<FORM>段中传递给服务器的用户请求部分的完整内容。

表 4-3　Request 对象的方法及说明

方　　法	说　　明
BinaryRead(count)	当数据作为 POST 请求的一部分发往服务器时，从客户请求中获得 count 字节的数据，返回一个 Variant 数组(或 SafeArray)。如果 ASP 代码已经引用了 Request.Form 集合，这个方法就不能用。同样，如果用了 BinaryRead 方法，就不能访问 Request.Form 集合

4.1.2　从浏览器端取得数据

Request 对象可以收集并处理用户通过 HTTP 请求传递的所有信息，包括 HTML 表格用 POST 方法或 GET 方法传递的参数、Cookie 数据和用户认证等。Request 的语法结构如下：

```
Request [. 集合 | 属性 | 方法 ]( 变量 )
```

Request 对象包含 3 类成员，分别为集合、属性和方法，其中集合包含了客户端的数据内容，如表 4-4 所示列出了 Request 对象的集合成员。Request 对象的属性与方法各有一个，并在 ASP 网页中很少使用，所以这里不再说明。

表 4-4　Request 对象的集合成员

集　合	说　明
Cookies	允许用户检索在 HTTP 请求中发送的 Cookie 的值
Form	当<form>标记的方法设为 POST，检索 HTTP 请求正文中的表格元素的值
QueryString	检索 HTTP 查询字符串变量的值，HTTP 查询字符串由问号(？)后的值指定
ServerVariables	客户端对服务器提出请求，同时传送到服务器的 HTTP 标题与服务器变量等数据

例如，下面的语句用于从 HTML 表单中取得用 POST 方法传递的 UserName 数据：

```
<% strUserName =Request.Form("UserName")%>
```

Request 语句中的集合名称也可以省略，如上面的语句也可以写成以下的形式：

```
<% strUserName =Request ("UserName")%>
```

如果在 Request 对象中没有指定准确的集合名称，ASP 会自动搜索来确定数据的获取方法。搜索的顺序是 QueryString、Form、Cookies 和 ServerVariables，ASP 逐一检查是否有信息输入，如果有则返回获得的变量信息。

4.1.3　Request.QueryString 集合读取表单 GET 方法数据

网页中常采取表单的形式与访问者进行交互。用户在表单中输入信息后，单击“确定”或“提交”按钮即可将信息传送到服务器上。服务器可获取这些信息进行下一步的处理和操作。在 HTML 中常见的 FORM 语句的语法结构如下：

```
<Form    Action=处理程序的网址    Method=Get|Post    Name=该 FORM 的名称>
……
</Form>
```

其中，Action 属性用于指定表单处理程序的 URL；Method 属性则指定提供数据的方法，可取值为 GET 与 POST 方式中的一个。表单选用 GET 方式时，ASP 要使用 Request.QueryString 集合来读取表单数据；选用 POST 方式时，ASP 则使用 Request.Form 集合来读取表单的数据。

当 HTML 表单用 GET 方法向 ASP 文件传递数据时，表单提交的数据不是被当作一个单独的包发送，而是被附在 URL 的查询字符串中一起被提交到服务器端指定的文件中。

QueryString 集合的功能就是从查询字符串中读取用户提交的数据。例如从下面的 URL 地址信息中，Request.QueryString 可得到 strName 和 Title 两个变量的值。

```
http://zhangshihua/4-1-login.asp?strName=赵刚&Title=Mr
```

注意：

查询字符串以问号(？)开始，包含几对字段名和分配的值。不同的字段名和值对用&符号连接。

【练习 4-1】使用 GET 方法传递图 4-1 所示的 HTML 表单中的值，并将结果在 ASP 页面上显示出来，显示结果如图 4-2 所示。

(1) 创建一个名为 QueryString.asp 的网页，其代码如下所示：

```
<Form Action="login1.asp" Method="Get" Name="login">
用户名：<Input Type=text Name=strName >
<Br><Br>
性  别：<Input Name="Title" Type="Radio" Value="Mr" Checked>先生
<Input Name="Title" Type="Radio" Value="Ms">女士
<Br><Br>
<Input Type="Submit" Value="确认提交">
<Input Type="Reset" Value="重新输入">
</Form>
```

(2) 将 QueryString.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 创建表单信息结果的程序文件 login1.asp，其代码如下所示：

```
<%
Dim strName,strTitle
strName=Request.QueryString("strName")
Response.Write"您的用户名为"&strName
Response.Write"<br>"
strTitle=Request.QueryString("Title")
If strTitle="Mr" Then
Response.Write"先生，您好！"
ELse
Response.Write"女士，您好！"
End If
%>
```

(4) 将 login1.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(5) 启动 IE 浏览器后，在地址栏中输入 http://localhost/QueryString.asp 并按 Enter 键，效果如图 4-1 所示。

(6) 在 QueryString.asp 页面的“用户名”文本框中输入用户的名称，如“杜思明”，在“性别”选项区域中选中“先生”单选按钮，然后单击“确认提交”按钮，浏览器将会打开如图 4-2 所示的 login1.asp 页面显示 QueryString.asp 页面中的输入结果。

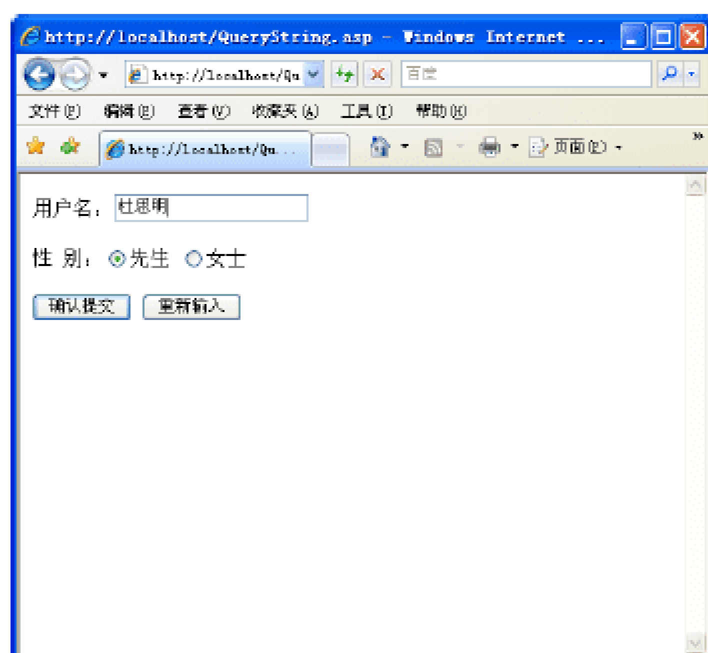


图 4-1　QueryString.asp 页面效果

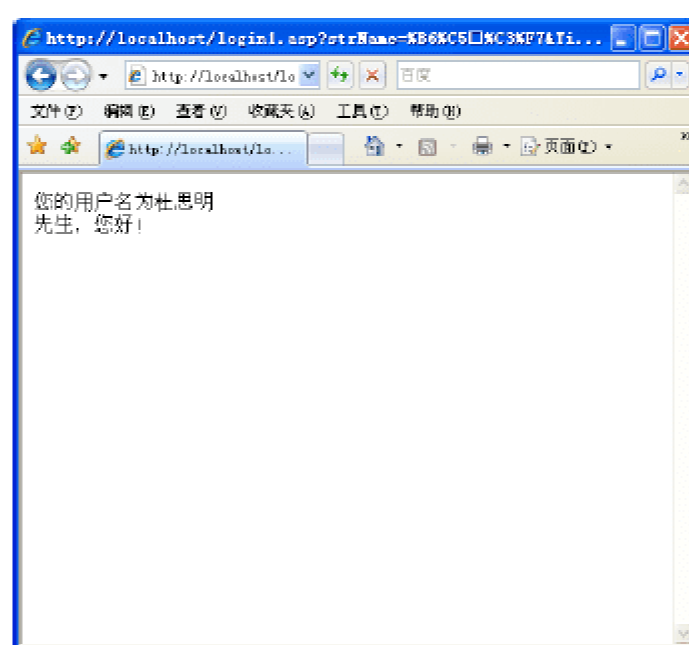


图 4-2　login1.asp 页面效果

4.1.4　Request.Form 集合读取表单 POST 方法数据

GET 方法的优点是可以方便地为服务器端传递信息，缺点是不能传递长而复杂的数据到服务器端，否则会造成数据的丢失，这是因为某些服务器会限制 URL 查询字符串的长度。因此，如果要将表单中的大量数据发送到服务器，应使用 POST 方法。

POST 方法在 HTTP 请求体内发送数据，几乎不限制发送到 Web 服务器的数据长度。检索使用 POST 方法发送的数据通常采用 Request 对象的 Form 集合来进行。

注意：

Form 集合和 Form 表单的区别：Form 表单是 HTML 提供的表单，并不是 ASP 特有的；Form 集合是特指 ASP 的 Request 对象获取信息的一种方法。两者的联系就是 ASP 用 Form 集合来获取 Form 表单中的数据信息。

【练习 4-2】编写一个计算器程序，可使用 POST 方法接受 HTML 表单中的数值和运算符，并将正确的计算结果在当前页面上显示出来，结果如图 4-3 所示。

(1) 创建一个名为 calculator.asp 的网页，其代码如下所示：

```
<Form Action=calculator.asp Method=post>
操作数 1: <Input Type=text Name=num1><br>
操作数 2: <Input Type=text Name=num2><br>
<p>
选择你要进行的操作<br>
<Input Type=radio Name=operation Value="加" checked>加<br>
<Input Type=radio Name=operation Value="减">减<br>
<Input Type=radio Name=operation Value="乘">乘<br>
<Input Type=radio Name=operation Value="除">除<br>
<Input Type=Submit><Input Type=Reset>
</Form>
<Hr>
<%
Dim n1,n2,op
```

```
If Request.form.count=0 Then
' Request.form.count 确定参数中值的个数。如果参数未关联多个值，则计数为 1
' 如果找不到参数，计数为 0
 Response.end
End If
n1=Request.form("num1")
n2=Request.form("num2")
op=Request.form("operation")
If op="加" Then
      Response.Write n1&"+"&n2&"="&clng(n1)+clng(n2)
Elseif op="减" Then
      Response.Write n1&"-"&n2&"="&clng(n1)-clng(n2)
Elseif op="乘" Then
      Response.Write n1&"*"&n2&"="&clng(n1)*clng(n2)
Elseif op="除" Then
      Response.Write n1&"/"&n2&"="&clng(n1)/clng(n2)
End If
%>
```

(2) 将 calculator.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/calculator.asp 并按 Enter 键，效果如图 4-4 所示。

(4) 在 calculator.asp 页面的“操作数 1”和“操作数 2”文本框中输入任意数字，并在“选择你要进行的操作”选项区域中选中所需的操作单选按钮，然后单击“提交查询内容”按钮，将会在页面的底部得到相应的计算结果，如图 4-3 所示。

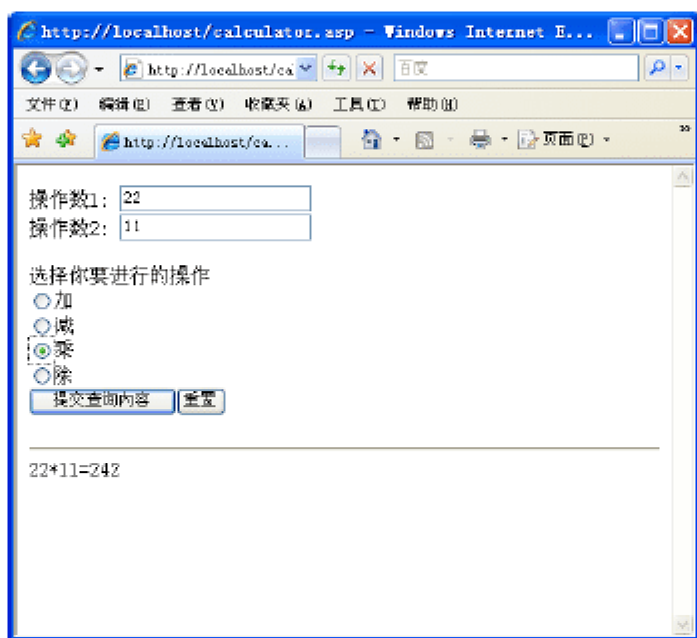


图 4-3　计算结果

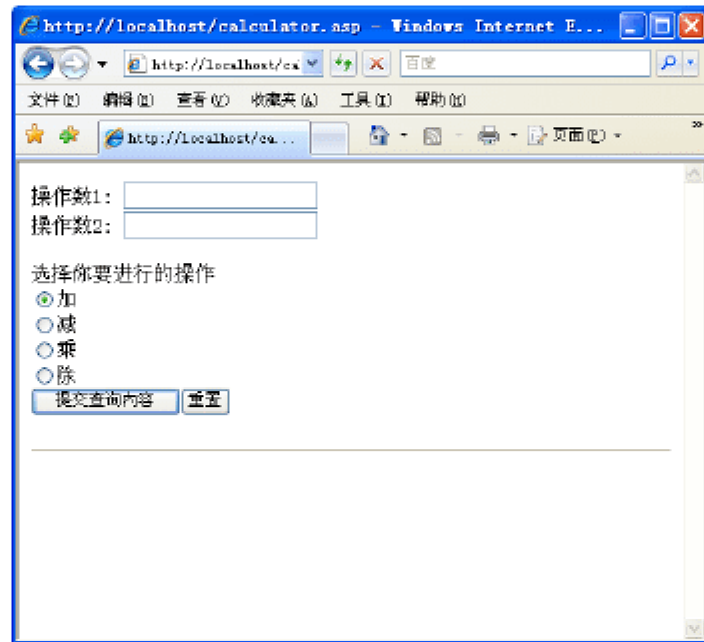


图 4-4　calculator.asp 页面效果

4.1.5　Request.ServerVariables 集合读取服务器端环境变量

在浏览器中浏览网页时使用的传输协议是 HTTP，在 HTTP 的标题文件中会记录一些客户端的信息，如客户的 IP 地址、浏览器版本及端口号等。

有时服务器端需要根据不同的客户端信息做出不同的反应，这时就需要用 Server

Variables 集合获取所需信息。常用的环境变量及其说明如表 4-5 所示。

表 4-5　常用的环境变量

变　　量	说　　明
CONTENT_LENGTH	客户端所提交内容的长度
CONTENT_TYPE	客户端所提交内容的类型，如 text/html，同附加信息的查询一起使用，如 HTTP 查询 GET、POST 和 PUT
HTTP_ACCEPT_LANGUAGE	浏览器用该变量向服务器传送用于显示内容所使用的语言，格式是 LA-CO 或 LA，LA 是语言的缩写，CO 是国家的缩写。若接受多种语言，则用逗号分隔开
HTTP_USER_AGENT	该变量包含浏览器的名字、版本和平台
HTTP_REFERER	确定哪个 Web 页调用脚本
LOCAL_ADDR	返回接受请求的服务器地址。如果在绑定多个 IP 地址的多宿主机上查找请求所使用的地址时，这个变量非常重要
PATH_INFO	客户端的路径信息。可以通过虚拟路径和 PATH_INFO 变量来访问脚本
QUERY_STRING	查询 HTTP 请求中问号(?)后的信息
REMOTE_ADDR	发出请求的远程主机的 IP 地址
REMOTE_HOST	发出请求的主机名称。如果服务器无此信息，它将 REMOTE_ADDR 变量设置为空
REQUEST_METHOD	该方法用于提出请求。相当于 HTTP 的 GET、HEAD、POST 等
SERVER_NAME	出现在自引用 URL 中的服务器主机名、DNS 域名或 IP 地址
SERVER_PORT	发送请求的端口号

列举 ServerVariables 集合包含的所有成员及其值的方法如下。

```
<OL>
<% For Each Key in Request.ServerVariables%>
<Li>
<B><%=Key%>=</B><%=Request.ServerVariables(key)%><Br>
<%Next%>
</OL>
```

根据对 ServerVariables 集合获取的信息进行处理，服务器端可对不同的客户端信息做出不同的反应。例如，如果某一计算机上安装了多个 Web 服务器，也就是有多个域名同时对应着一个 IP 地址，这时可以使用 SERVER_NAME 属性来根据不同的域名显示不同的页面。

【练习 4-3】读取环境变量 SERVER_NAME，然后根据检索到的域名来执行重定向。

(1) 创建一个名为 Cdx.asp 的网页，其代码如下所示：

```
<%
Response.Expires=0                  '设置页面的缓存过期时间为 0
Dim strServerName
strServerName=Request.ServerVariables("SERVER_NAME")
'此例需有一定的服务器配置环境才能看到实际的效果
Select Case strServerName
   Case "www.test1.com"             '当访问的域名为 www.test1.com 时
 Response.Redirect "test1.asp"      '重定向到 test1.asp 文件
   Case "www.test2.com"             '当访问的域名为 www.test2.com 时
 Response.Redirect "test2.asp"      '重定向到 test2.asp 文件
   Case Else
 Response.Redirect "notFound.asp"
End Select
%>
```

(2) 将 Cdx.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Cdx.asp 并按 Enter 键测试网页。这时，ASP 网页会根据读取的环境变量 SERVER_NAME，打开不同的网页(当服务器访问 www.test1.com 网站时，Cdx.asp 将自动打开 test1.asp 页面，当服务器访问 www.test2.com 网站时，则打开 test2.asp 页面，如果服务器没有打开任何一个网站，Cdx.asp 页面将重定向到 notFound.asp 页面)。

4.2　Response 对 象

Response 对象用于向客户浏览器发送服务器端的数据。用户可以使用该对象将服务器端的数据用 HTML 的格式发送到客户端浏览器。该对象的功能与 Request 对象的功能相反，Request 对象用于得到用户提交的数据，而 Response 对象用于将服务器端的数据发送到用户浏览器，这是实现网页动态效果的基础。

4.2.1　Response 对象简介

Response 对象用于控制发送给用户的信息，包括直接发送信息给客户端浏览器，重定向浏览器到另外一个 URL 以及设置 Cookie 的值。

1. Response 对象的集合

如表 4-6 所示为 Response 对象的唯一集合，该集合设置希望放置在客户系统上的 Cookie 的值，它直接等同于 Request.Cookies 集合。

表 4-6　Response 对象的集合及说明

集 合 名 称	说　　明
Cookies	在当前响应中，返回客户端的所有 Cookie 的值，这个集合为只写的

2. Response 对象的属性

Response 对象也提供一系列的属性，可以读取和修改，使响应能够适应请求。当设置某些属性时，使用的语法可能与通常所使用的有一定的差异，如表 4-7 所示。

表 4-7　Response 对象的属性及说明

属　　性	说　　明
Buffer =True/False	读/写，布尔型，表明由一个 ASP 页面所创建的输出是否一直存放在 IIS 缓冲区，直到当前页面的所有服务器脚本处理完毕或 Flush、End 方法被调用。在任何输出(包括 HTTP 报头信息)送往 IIS 之前这个属性必须设置。因此在.asp 文件中，这个设置应该在<%@LANGUAGE=...%>语句后面的第一行。在 ASP 3.0 以及后续版本中默认设置缓冲为开(True)，而在早期版本中默认为关(False)
CacheControl "setting"	读/写，字符型，设置这个属性为 Public，允许代理服务器缓存页面，如果为 Private，则禁止代理服务器缓存的发生
Charset="value"	读/写，字符型，在由服务器为每个响应创建的 HTTP Content-Type 报头中附上所用的字符集名称(例如 ISO-LATIN-7)
Content Type="MIME-type"	读/写，字符型，指明响应的 HTTP 内容类型，标准的 MIME 类型(例如 text/xml 或 Image/gif)。假如省略，表示使用 MIME 类型 text/html，内容类型告诉浏览器所期望内容的类型
Expires minutes	读/写，数值型，指明页面有效的以分钟计算的时间长度，假如用户请求其有效期之前的相同页面，将直接读取显示缓冲中的内容，这个有效期过后，页面将不再保留在私有(用户)或公用(代理服务器)缓冲中
Expires Absolute #date[time]#	读/写，日期/时间型，指明当某页面过期和不再有效时的绝对日期和时间
IsClientConnected	只读，布尔型，返回客户是否仍然连接和下载页面的状态标志。在当前的页面已执行完毕之前，假如一个客户转移到另一个页面，这个标志可用来中止处理(使用 Response.End 方法)

(续表)

属　　性	说　　明
PICS "PICS-Label-string"	只写，字符型，创建一个 PICS 报头并将其加到响应中的 HTTP 报头中，PICS 报头定义页面内容中的词汇等级，如暴力、性、不良语言等
Status="Code message"	读/写，字符型，指明返回客户响应的 HTTP 报头中表明错误或页面处理是否成功的状态值和信息。例如 200 OK 和 404 Not Found

3. Response 对象的方法

如表 4-8 所示为 Response 对象提供一系列的方法，它允许直接处理为传给客户端而创建的页面内容。

表 4-8　Response 对象的方法及说明

方　　法	说　　明
AddHeader("name","content")	通过使用 name 和 content 值，创建一个定制的 HTTP 报头，并增加到响应中。不能替换现有的相同名称的报头。一旦增加了一个报头就不能删除。这个方法必须在任何页面内容(即 text 和 HTML)被发往客户端前使用
AppendToLog("string")	当使用 W3C Extended Log File Format 文件格式时，对于用户请求的 Web 服务器的日志文件增加一个条目。至少要求在包含页面的站点的 Extended Properties 中选择 URIStem
BinaryWrite(SafeArray)	在当前的 HTTP 输出流中写入 Variant 类型的 SafeArray，而不经过任何字符转换。对于写入非字符串的信息，例如定制的应用程序请求的二进制数据或组成图像文件的二进制字节，是非常有用的
Clear()	当 Response.Buffer 为 True 时，从 IIS 响应缓冲中删除现存的缓冲页面内容。但不删除 HTTP 响应的报头，可用来放弃部分完成的页面
End()	让 ASP 结束处理页面的脚本，并返回当前已创建的内容，然后放弃页面的进一步处理
Flush()	发送 IIS 缓冲中所有当前缓冲页面给客户端。当 Response.buffer 为 True 时，可以用来发送较大页面的部分内容给个别的用户
Redirect("url")	通过在响应中发送一个 302 Object MovedHTTP 报头，指示浏览器根据字符串 URL 下载相应地址的页面
Write("string")	在当前的 HTTP 响应信息流和 IIS 缓冲区写入指定的字符，使之成为返回页面的一部分

4.2.2　利用缓冲区输出数据

使用浏览器打开一个网页时，有时需要等待很长的一段时间，这是因为设置了页面缓冲输出。页面缓冲利用 Response 对象在缓冲区输出数据。

1. Buffer 属性

在 ASP 程序中，可以为页面在服务器端设置一个缓存区。缓存区是一个存储区，可以在其释放数据之前容纳一段时间，缓冲区的优点在于它的行为可以进行控制。

设置缓存后，服务器端可减少与客户端连接的次数而提高整体的响应速度，并可在满足某些条件(如脚本处理不正确或用户没有适当的安全证书)时撤销已经处理的结果，而不会出现响应完成一部分就停止的状况。

缓存功能的打开和关闭是通过 Response 对象的 Buffer 属性来完成的。若将 Buffer 属性设为 False，则关闭缓存功能，Web 服务器在处理页面时会随时返回 HTML 和脚本结果；若将 Buffer 属性设为 True，则打开缓存功能，Web 服务器在处理页面时会将结果暂时存放到缓存中，当全部脚本处理完后，或遇到 End 或 Flush 方法时，才将缓存中的内容发送到浏览器。

Buffer 属性的更改必须放在 HTML 或脚本输出之前。这是因为在任何内容发送到浏览器后，Buffer 属性值就不能再更改，否则会引起错误。

注意：

如果在 ASP 文件中的任意地方用到 Redirect 方法重定向页面，必须在文件开头关闭 Buffer 属性，否则就会报错。

【练习 4-4】创建一个用于测试 Buffer 属性的 ASP 应用程序，该程序会在浏览器上显示 1~255 之间的数字，并且每输出一个数字就自动换 1 行。

(1) 创建一个名为 Buffer.asp 的网页，其代码如下所示(如图 4-5 所示)：

```
<html>
<head> 1=255
<title>测试 buffer 属性的例子</title>
</head>
<body>
<%
For i=1 to 255                    'i 为 1~255 之间的一个值
Response.write(i&"<BR>")          '输出 i 的值并换行
next
%>
</body>
```

(2) 将 Buffer.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器，在地址栏中输入 http://localhost/Buffer.asp 并按 Enter 键，效果如图 4-6 所示。



图 4-5　Buffer.asp 的代码

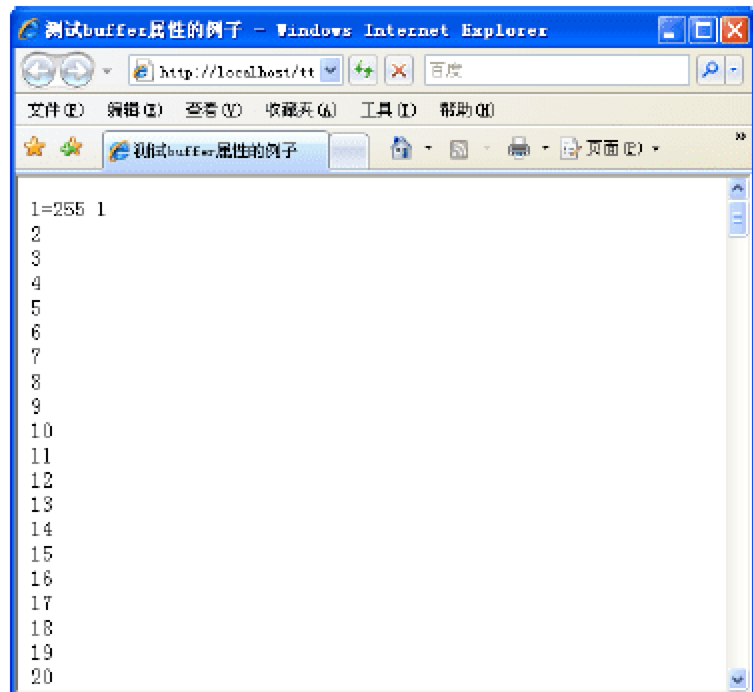


图 4-6　Buffer.asp 的效果

注意：

本节实例创建的程序脚本在屏幕上显示 1~255 之间的数字，并且每输出一个数字就换一行。每执行一句命令后都会立即显示结果。如果把 Buffer 属性设置为 True，那么服务器端的 Response 要写入缓存区中，当脚本处理完成后再释放给用户。如果把 Buffer 属性设置为 False，则在服务器处理脚本时，HTML 要顺序地发给客户程序(上面实例的 Buffer 属性使用默认值 False)。

2. ContentType 属性

ContentType 属性指定响应的 HTTP 内容类型，其语法结构如下：

```
Response.ContentType  [=ContentType]
```

ContentType 字符串通常被格式化为类型/子类型，其中类型是常规内容范畴，子类为特定内容类型。如果未指定 ContentType，默认为 text/HTML。

Web 服务器将某个文件发送到浏览器时，将文件的 MIME 类型告诉浏览器，浏览器根据文件的 MIME 类型和扩展名来确定是自己本身就能显示，还是必须调用其他应用程序。

注意：

MIME 即 Multipurpose Internet Mail Extensions(多用途 Internet 邮件扩展)，作为对 SMTP 协议的扩充，MIME 规定了通过 SMTP 协议传输非文本电子邮件附件的标准。目前，MIME 的用途早已经超越了收发电子邮件的范围，成为在 Internet 上传输多媒体信息的基本协议之一，是 Internet 上识别文件类型的标准方法。

3. Expires 属性

Expires 属性指定了在浏览器上缓冲存储的页距过期还有多少时间。如果用户在某个页过期之前又回到此页，就会显示缓冲区中的版本。其语法结构如下：

```
Response.Expires [= 时间]
```

时间参数设置网页距过期还有多少分钟。如果将此参数设置为 0，可使缓存的页立即过期，这样客户端每次都将从服务器上得到最新的页面。

4. Expires Absolute 属性

Expires Absolute 属性指定缓存于浏览器中的页面的确切到期日期和时间。在未到期之前，若用户返回到该页，就显示该缓存的页。如果未指定时间，该主页在当天午夜到期。如果未指定日期，则该主页在脚本运行当天的指定时间到期。其语法结构如下：

```
Response.ExpiresAbsolute [= [日期] [时间] ]
```

例如，以下代码指定页面在 2007 年 7 月 31 日 16:17 分 15 秒到期。

```
<% Response.ExpiresAbsolute=#July 31,2007 16:17:15# %>
```

如果 Expires 属性和 Expires Absolute 属性在一个页面上设置了多次，ASP 会自动选择使用最短的设置时间。

5. Status 属性

Status 属性用来传递服务器 HTTP 响应的状态。这个属性可以用来处理 HTTP 请求后服务器返回的错误。

服务器返回的状态码由 3 位数字构成，可以根据状态码来确定服务器是如何处理 HTTP 请求的。在调试过程或向客户端返回有关错误信息时，Status 属性特别重要。如表 4-9 所示对常见的状态码做了说明。

表 4-9　HTTP 响应状态码

码	状　　态	码	状　　态
200	OK	401	Unauthorized
201	Created	403	Forbidden
202	Accepted	404	Page not found
204	No content	500	Internet server error
301	Moved permanently	501	Not implemented
302	Moved temporarialy	502	Bad gateway
304	Not modified	503	Service unavailable
400	Bad request		

6. Charset 属性

Charset 属性将字符集名称附加到 Response 对象中的 ContentType 标题后面，其语法结构如下：

```
Response.Charset (字符集)
```

对于不包含 Response.Charset 属性的 ASP 页，Contenttype 标题将为 content-type:text/html，如果 ASP 文件包含了下面一句脚本：

```
<% Response.Charset("ISO-LATIN-7") %>
```

则 Contenttype 标题将为 content-type:text/html;charset=ISO-LATIN-7。

注意：

无论字符串表示的字符集是否有效，该功能都会将其插入 Contenttype 标题中。如果某个页包含多个含有 Response.Charset 的标记，则每个 Response.Charset 都将替代前一个 CharsetName。这样，字符集将被设置为该页中 Response.Charset 的最后一个实例所指定的值。

7. IsClientConnected 属性

IsClientConnected 属性是只读属性，用来指示自上次调用 Response.Write 后，客户端是否与服务器相连。其语法结构如下：

```
Response.IsClientConnected ( )
```

IsClientConnected 属性允许用户在客户端与服务器没有连接的情况下有更多的控制。例如，在从客户端提出请求起到服务器作出响应，其间要用去很长的一段时间，IsClientConnected 属性就可以确保在继续处理脚本之前客户端仍是连通的。

8. CacheControl 属性

CacheControl 属性用来控制是否允许代理服务器高速缓存。CacheControl 属性的默认值为 Private，可以阻止代理服务器高速缓存页面信息。当属性值为 Public 时，代理服务器可以缓冲由 ASP 产生的输出。其语法结构如下：

```
Response.CacheControl [=缓冲存储器控制标题 ]
```

【练习 4-5】设置网页的缓存时间并控制在浏览器的显示内容。

(1) 创建一个名为 CacheControl.asp 的网页，其代码如下所示：

```
<%
Response.Buffer=TRUE        '将此页面设为缓存页，服务器暂时不将显示内容发送到浏览器
Response.Expires=60         '设置缓存过期时间为 60 分钟
%>
设置为缓存时间为一个小时的第一段测试的语句。<Br>
<%
Response.Clear              '清除缓存中的所有内容
Response.Expires=0          '设置缓存过期时间为 0 分钟
%>
```

```
第二段会被显示的测试语句。
<%
Response.Flush                '将缓存区的内容立即传送到浏览器
%>
<%
Response.End                  'Web 服务器停止当前的脚本处理并返回当前结果
%>
不会被送往浏览器显示的第三段测试语句。
```

(2) 将 CacheControl.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

4.2.3　实现网页的重定位

本章的第 4.2.2 节介绍了 Response 对象的属性，本节将重点介绍 Response 对象的方法。在 ASP 应用程序中，使用 Response 对象的方法可以根据客户端不同的请求输出不同的返回结果。

1. Write 方法

Write 方法是 Response 对象中最常用的方法之一，它可以把变量的值发送到用户端的当前页面。Write 方法的功能很强大，几乎可以输出所有的对象和数据。

在 Write 方法中可以嵌入任何 HTML 标记，只要该标记是合法的。如下例将输出绿色的汉字和一条水平线：

```
Response.Write "<Font color=green>"
Response.Write "欢迎您来访问！"&"<Br>"
Response.Write "</Font>"
Response.Write "<Hr>"
```

将 HTML 标记与 ASP 中的变量恰当地结合使用，可使程序更简洁易读。如下例是根据 Request 对象获取的数据来创建表格的一行信息：

```
<%Response.Write "<Tr><Td>"&Request.Form("strUserName")&"</Td><Td>"_
& Request.Form("intAge")&"</Td></Tr>" %
```

2. Clear 方法

Clear 方法用于清除缓冲区的所有 HTML 输出，但它只删除响应正文而不删除响应标题。在服务器上的程序产生错误时，可用 Clear 方法来处理错误情况。

注意：

Clear 方法仅当 Response 对象的 Buffer 属性设为 True 时才起作用。如果 Buffer 属性未设为 True，则 Clear 方法将导致运行错误。

3. Flush 方法

调用 Flush 方法时，缓存中的所有内容会立即发送到客户端。与 Clear 方法一样，如果 Response 对象的 Buffer 属性没有设为 True，则 Flush 方法同样会产生运行错误。

4. End 方法

End 方法使服务器停止当前脚本的处理并返回当前结果。如果 Response 对象的 Buffer 属性设为 True，则 End 方法立即把缓存中的内容发送到客户端并清除缓存。因此，若想取消向客户端的所有输出，可以先用 Clear 方法清除缓存，再用 End 方法停止脚本的处理。

5. Redirect 方法

在普通网页中，可以使用超链接的方式来引导访问者跳转到另一个页面，但这个过程需要访问者单击一个超链接才可以进行。Response 对象的 Redirect 方法则可以自动完成页面间的跳转，而访问者几乎不会感觉出来。

【练习 4-6】创建一个搜索页面，如图 4-7 所示。使用 Response.Redirect 方法编写程序来完成不同搜索引擎的转向定位，搜索结果如图 4-8 所示。

(1) 创建一个名为 Redirect.asp 的网页，其代码如下所示：

```
<Form Method="get" Action="redirect.asp">
<Input Type="Text" Name="SearchStrings">
<Input Type="Submit" Name="Search" Value="搜索">
<p>
<Input Name="goURL" Type="Radio" Value="百度" Checked>百度
<Input Name="goURL" Type="Radio" Value="Yahoo">Yahoo
</Form>
<%
Response.Buffer=True                                          '打开缓存功能
Dim strUrlRedirTo,strSearchStrings
strSearchStrings=Trim(Request.QueryString("SearchStrings"))   '获取欲搜索的关键字
If (Len(strSearchStrings)) Then
strSearchStrings=Server.UrlEncode(strSearchStrings)     '对字符串按 URL 规则进行编码
If Request.QueryString("goURL")="百度"   Then
strUrlRedirTo="http://www.baidu.com/baidu?word="&strSearchStrings     '产生跳转的 URL
End if
If Request.QueryString("goURL")="Yahoo"   Then
strUrlRedirTo="http://cn.websearch.yahoo.com/search/web_cn?p="&strSearchStrings
End If
Response.Redirect strUrlRedirTo                    '跳转语句
End If
%>
```

(2) 将 Redirect.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Redirect.asp 并按 Enter 键，效果如图 4-7 所示。

(4) 在图 4-7 所示页面的文本框中输入 ASP，并选中“百度”单选按钮，然后单击“搜索”按钮，ASP 页面将自动打开如图 4-8 所示的页面，通过百度搜索引擎搜索关键字 ASP。

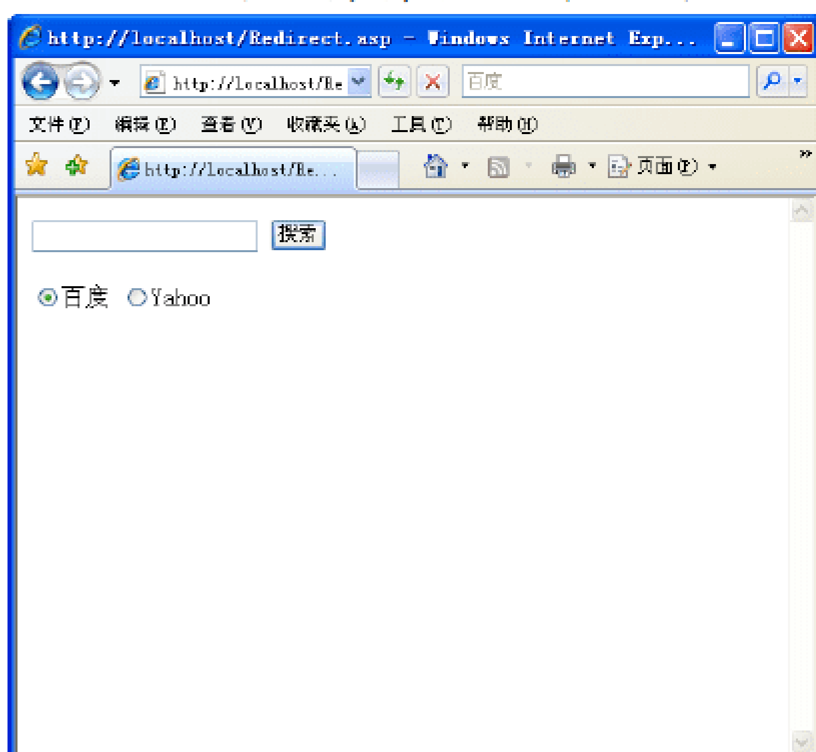


图 4-7　Redirect.asp 页面效果



图 4-8　搜索结果

6. Binary Write 方法

BinaryWrite 方法不经任何字符转换就将指定的信息写到 HTTP 输出。该方法用于写非字符串信息，如客户端应用程序所需的二进制数据，常见的有图形、声音或影像等。其语法结构如下：

```
Response.BinaryWrite　输出数据
```

如果有一个产生字节数组的对象，就可调用 BinaryWrite 将这些生成的字节发送给客户端应用程序，如下例所示：

```
<%
Set BinGen = Server.CreateObject(MY.BinaryGenerator)
Pict = BinGen.MakePicture
Response.BinaryWrite Pict
%>
```

7. AddHeader 方法

AddHeader 方法用指定的值添加 HTML 标题。该方法常常向响应添加新的 HTTP 标题，并不替代现有的同名标题。一旦标题被添加，将不能删除。其语法结构如下：

```
Response.AddHeader 标题变量名称，初始值
```

为避免命名不明确，标题变量名称中不能包含任何下划线字符“_”。由于 HTTP 协议要求所有的标题都必须在内容之前发送，所以必须在任何输出(例如由 HTML 或 Write 方法

生成的输出)发送到客户端之前在脚本中调用 AddHeader。但当 Buffer 属性被设置为 True 时例外。若输出被缓冲，则可以在脚本中的任何地方调用 AddHeader 方法，只要它在 Flush 之前执行即可。

8. AppendToLog 方法

AppendToLog 方法在 Web 服务器日志文件的末尾增加一项，每项内容最多 80 个字符。可以在脚本的同一部分多次调用该方法。每次调用该方法时，都会在当前条目中添加指定的字符串。其语法结构如下：

```
Response.AppendToLog 字符串
```

输出的字符串为要添加到日志文件中的文本。由于日志文件的每项是以逗号分隔的，所以字符串的内容中不能含有逗号，字符串最大长度为 80 个字符。

注意：

服务器日志文件是记载访问情况的纯文本文件，默认放置于 Windows/System32/LogFiles 文件夹中，可以使用任何文本编辑器来读写。

4.3　在网页中使用 Cookie

Cookie 是一种标记，由 Web 服务器嵌入用户浏览器中来标识用户。当下次同一个浏览器请求网页时，将把以前从 Web 服务器得到的 Cookie 再传送给服务器。Cookie 允许一个用户关联一组信息。Cookie 被保存为简单的文本文件，其名称标识用户和站点，可以用任何文本编辑器打开。

4.3.1　认识 Cookie

Cookie 是 Web 服务器嵌入客户机中以标识用户的标记，是服务器或脚本可以维护用户信息的一种方式。

当用户访问一个 Web 站点时，Web 服务器可以在用户的计算机上建立一个 Cookie 文件，里面包含用户的有关信息，如身份识别号码、密码、用户在 Web 站点上购物的方式或用户访问该站点的次数。以后，无论何时连接到 Web 站点服务器，都可以访问这个 Cookie 文件，来查询和修改其中的信息。

每个网站都可以设置自己的 Cookie，Cookie 中的数据完全由 Web 服务器的管理者来决定。例如，当 Web 服务器想知道某个用户是否曾进入到它的站点或想获得某个使用者的相关资料时，便可使用 Cookie。

Cookie 有会话 Cookie 和永久 Cookie 两种形式。前者是临时的，只在浏览器打开时存

在；后者是永久的，存在于用户的硬盘上并在指定日期过期之前一直可用。

4.3.2 Cookie 的属性

Cookie 的属性用于 Web 服务器指定欲设置的 Cookie 值，各种属性的名称及说明如表 4-10 所示。

表 4-10 Cookie 的属性

属 性	说 明
Expires	只写。指定 Cookie 的过期日期。要在会话结束后将 Cookie 存在用户的硬盘上，必须设置该属性。过了该属性设置的日期后，Cookie 就不能使用了。通过给 Cookie 赋一个过期的日期，就可以删除 Cookie。如果不对 Expires 属性赋值，那么默认用户一离开网站 Cookie 就过期
Domain	只写。若指定，则 Cookie 只被发送到对指定域的请求中去
Path	只写。若指定，则 Cookie 只被发送到对指定路径的请求中去。若未设置该属性，则使用应用程序的路径
HasKeys	只读。确定 Cookie 是否是一个具有多个键值的 Cookie 字典，若是，则返回 True
Secure	只写。确定 Cookie 是否是安全的，Secure 属性设为 True，则 Cookie 传递中就实行了加密算法

4.3.3 设置 Cookie

ASP 脚本可用 Response 和 Request 对象的 Cookies 集合来设置和获取 Cookies 的值。创建 Cookie 的方法有两种：一种是创建单值的 Cookie；另一种是创建 Cookie 字典类型的 Cookie，即 Cookie 允许存在多个键值对。

1. 创建单值 Cookie

Response 对象的 Cookies 集合用来设置 Cookie，如下例所示：

```
<%
Response.Cookies("Username")="Zhangshihua"
Response.Cookies("Username").Expires="July 22,2007"
%>
```

本例将名字为 Username 的 Cookie 赋值为 Zhangshihua，同时将过期时间限定为 2007 年 7 月 22 日。

如果该 Cookie 名称不存在，则浏览器会自动新建一个名称并执行赋值操作。如果该 Cookie 已存在，浏览器会重新给 Cookie 赋值以覆盖原值。

注意：

由于 Cookie 是作为 HTTP 的 Header 部分来发送给客户端，那么该语句必须放在 ASP 的任何输出语句之前执行，或使用 Buffer 输出。

2. 创建多键值 Cookie

一个 Cookie 可以有多个键值，这样的 Cookie 称为 Cookie 字典，一个 Cookie 字典可以含有多个键值对，如下例所示：

```
<%
Response.Cookies("User")("Name")= "Zhangshihua"
Response.Cookies("User")("Sex")= "male"
Response.Cookies("User")("Password")= "21292390"
Response.Cookies("User").Expires="July 22,2004"
%>
```

上面的代码创建了一个名为 User 的 Cookie 字典，其中含有 3 个键，分别是 Name、Sex 和 Password，这 3 个键和相应的值都在一个大的 Cookie 中。

注意：

在 Response 对象的 Cookies 集合中定义多个名称，就可以为一个网站创建多个 Cookie 了。但是，许多浏览器对一个指定网站就限定了 3~4 个 Cookie。利用 Cookie 技术传递诸如密码这样的信息要特别小心。如果网站有安全接口层技术，也可以进行加密传输，但是在浏览器端该信息还是存放在文本文件中。

使用 HasKeys 属性来判断一个 Cookie 是否是一个 Cookie 字典，可以用如下的代码：

```
<%=Request.Cookies("名称").HasKeys %>
```

如果返回值为 True，那么它就是一个 Cookie 字典。

4.3.4　输出 Cookie 中保存的值

Request 对象的 Cookies 集合用来输出 Cookie 值，如下例所示：

```
<%=Request.Cookies("Username") %>
```

此例将输出名为 Username 的 Cookie 值。另外，可用 For Each 循环或 Count 属性和 For…Next 循环结合的方式来将 Cookie 集合的所有属性值显示出来，如下例所示：

```
<%
For EACH thing IN Request.Cookies
Response.write("<BR>"&thing&Request.Cookies(thing))
NEXT
```

```
%>
```

对于 Cookie 字典，Response 对象的 Cookies 集合可以按每一个键的相应名称显示相应键值，如下例所示：

```
<%=Request.Cookies("User")("Name")%>
<%=Request.Cookies("User")("Sex")%>
```

4.4　习　　题

4.4.1　填空题

(1) ___________对象用于接收客户端浏览器提交的数据，而___________对象的功能则是将服务器端的数据发送到客户端浏览器。

(2) 如果在 Request 对象中没有指定准确的集合名称，ASP 会自动按 QueryString、___________、___________和 ServerVariables 的顺序来搜索确定数据的获取方法。

(3) 当 HTML 表单用___________方法向 ASP 文件传递数据时，用户提交的数据将附在 URL 的查询字符串中一起被提交到服务器端指定的文件中。

(4) Response 的___________方法则可以自动完成页面间的跳转。

(5) 缓存功能的打开和关闭是通过 Response 对象的___________属性来完成的。

4.4.2　选择题

(1) Response 对象的(　　)属性可指定缓存页面的确切到期日期和时间。

A. Buffer 属性　　B. Expires 属性

C. ExpiresAbsolute 属性　　D. CacheControl 属性

(2) 下面不属于 Request 对象的集合成员的是(　　)。

A. Cookies　　B. Form

C. QueryString　　D. Server

(3) 下面不属于 Request.ServerVariables 集合可以读取服务器端环境变量的是(　　)。

A. 客户端的网速

B. 客户端所提交内容的数据类型

C. 发送请求的端口号

D. 发出请求的远程主机的 IP 地址

4.4.3　问答题

(1) 简述什么是对象。

(2) 简述 Request 对象和 Response 对象的作用及相互关系。

(3) 简述什么是 Cookie，有什么作用。

4.4.4　操作题

(1) 参照本章练习，编辑 ASP 程序以达到如图 4-9 所示的网页效果。

图 4-9　网页效果 1

(2) 参照本章所介绍的内容，编辑 ASP 程序以达到如图 4-10 所示的网页效果。

图 4-10　网页效果 2

第5章　Server 对 象

Server 对象主要用于服务器的相关操作，例如建立对象，对字符串应用 URL 或 HTML 编码，将虚拟目录映射到物理路径及设置脚本超时等。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 了解 Server 对象的属性
- 掌握 CreateObject 方法
- 掌握 URLEncode 方法
- 掌握 Transfer 方法
- 掌握 Execute 方法

5.1　认识 Server 对象

Server 对象提供对服务器上的方法和属性的访问，其中大多数方法和属性是为实用程序的功能服务的，如表 5-1 所示。

表 5-1　Server 对象成员

方　法	说　明
CreateObject(objName)	创建对象实体
HTMLEncode(string)	HTML 字符串编码
URLEncode(string)	路径字符串编码
MapPath(urlString)	取得绝对路径
Transfer(urlString)	转向至指定浏览网页
Execute(urlString)	执行外部网页
GetLastError()	取得 Error 对象

在这些成员中，使用最为频繁的方法是 CreateObject，它使网页可以创建一个指定的对象，同时利用这个对象进行所需的相关操作。例如，制作数据库的功能首先要使用 CreateObject 方法创建所需的 ADO 对象，再创建其他的方法包含改变网页的文字输出格式，获取网页路径等。

5.2　Server 对象的属性

Server 对象只有一个 ScriptTimeout 属性，用于指定一个脚本延时的时间期限。脚本运行超过 ScriptTimeout 属性设置时间将作超时处理，中止没有完毕的响应并提示超时错误信息。

可以在 IIS 的 Internet 信息服务程序中为 Web 服务设置默认的脚本超时值。如果将默认值设为－1，则脚本将永不会过期。如果设置的 ScriptTimeout 属性值小于默认值，则以默认值为超时上限。可以使用下面的语句来取得脚本的超时值：

```
<% TimeOut=Server.ScriptTimeOut %>
```

对于一些脚本执行需时过长的特定页面，也可使用 ScriptTimeout 属性为其设定一个限制时间。

【练习 5-1】在网页中随机显示一个星号，查看限制时间对页面响应的限制作用，程序运行效果如图 5-1 所示。

(1) 创建一个名为 page.asp 的网页，其代码如下所示(如图 5-2 所示):

```
<%Server.ScriptTimeOut=150%>
<%
For k=1 to 10
   Nextsecond=dateAdd("s",10,time)
     Do While time<nextsecond
     loop
     starx=starx+10*rnd()-1
     For i=1 to starx
     Response.Write (" ")
     Next
Response.Write ("*<p>")
Next
%>
```

(2) 将 page.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/page.asp 并按 Enter 键，效果如图 5-2 所示。

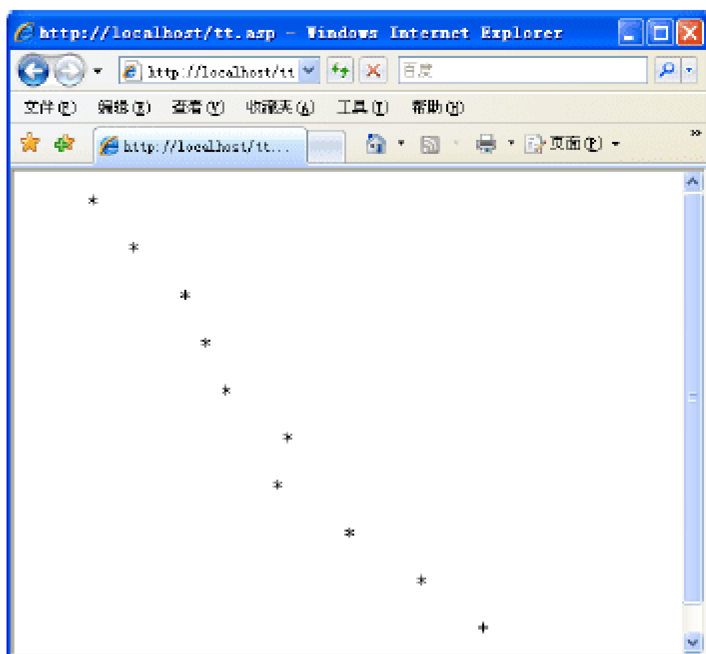


图 5-1　page.asp 的效果

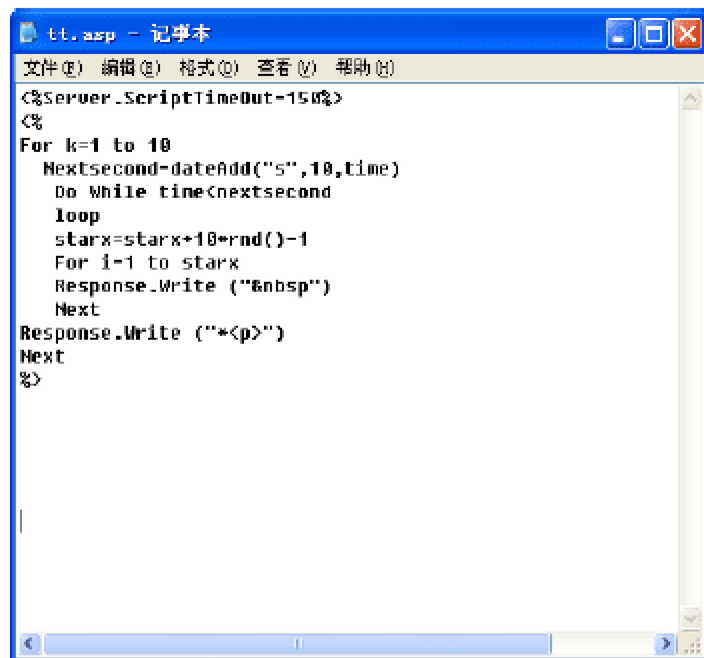


图 5-2　page.asp 的代码

5.3　Server 对象的方法

Server 对象最常用的方法是创建服务器组件的实例(Server.CreateObject)，其他方法用于将 URL 或 HTML 编码成字符串，将虚拟路径映射到物理路径，以及设置脚本的超时期限等。

5.3.1　CreateObject 方法

CreateObject 方法是 Server 对象的最重要方法，用于创建已注册到服务器上的组件的实例。

利用 Server 对象的 CreateObject 方法在网页中创建所要运用的对象实体，引用 CreateObject 的语法如下：

```
Set objCreate = Server.CreateObject("ObjectID")
```

其中，ObjectID 为所创建的对象识别名称，objCreate 为存储此对象的变量，一旦这行程序代码执行完成后，objCreate 便能够使用在该对象的网页之中，就如同 ASP 内建的对象一样，只是必须使用 objCreate 变量名称取代对象的识别 ID。

5.3.2　HTMLEncode 方法

浏览器通过解读 HTML 标记能显示各种格式的文本，但它却显示不了 HTML 标记本身。如下面的例子：

```
<% Response.Write("回车的 HTML 标记是：<Br>") %>
```

当执行这个语句时，浏览器会把
自动解释为回车命令，浏览器上显示的最终结果为：

```
回车的 HTML 标记是：
```

HTMLEncode 方法用以转换 HTML 标记，避免这些特定字符被浏览器进一步作解译。HTMLEncode 将 HTML 标记字符转换为不由浏览器解释的字符代码。如下面的例子：

```
<% Response.Write Server.HTMLEncode ("回车的 HTML 标记是：<Br>") %>
```

当执行这个语句时，浏览器上显示的最终结果为：

```
回车的 HTML 标记是：<Br>
```

【练习 5-2】利用 HTMLEncode 方法将一个指定的字符串按 HTML 的编码输出。

(1) 创建一个名为 HTML.asp 的网页，其代码如下所示(如图 5-3 所示):

```
<html>
<head>
<title>HTMLEncode</title>
```

```
</head>
<body bgcoloer="#ffffff">
<%
response.write"<b>hello</b><i>world</i>"
response.write"<p>"
response.write server.htmlencode("<b>hello</b><i>world</i>")
response.write"<p>"
response.write"Nanjing&Beijing"
response.write"<p>"
response.write server.URLencode("Nanjing and Beijing")
%>
</body>
</html>
```

(2) 将 HTML.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/HTML.asp 并按 Enter 键，效果如图 5-4 所示。

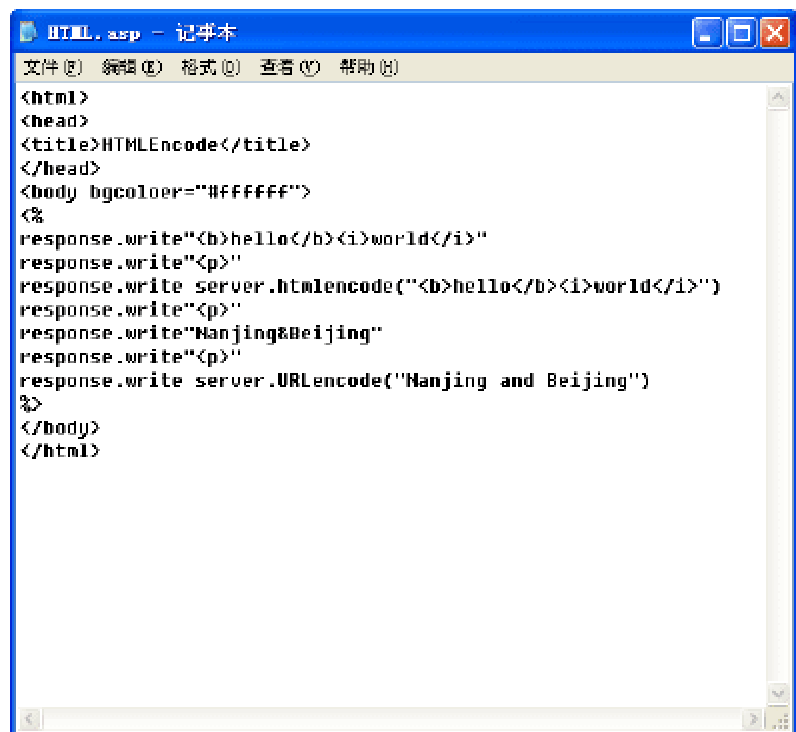


图 5-3　HTML.asp 的代码

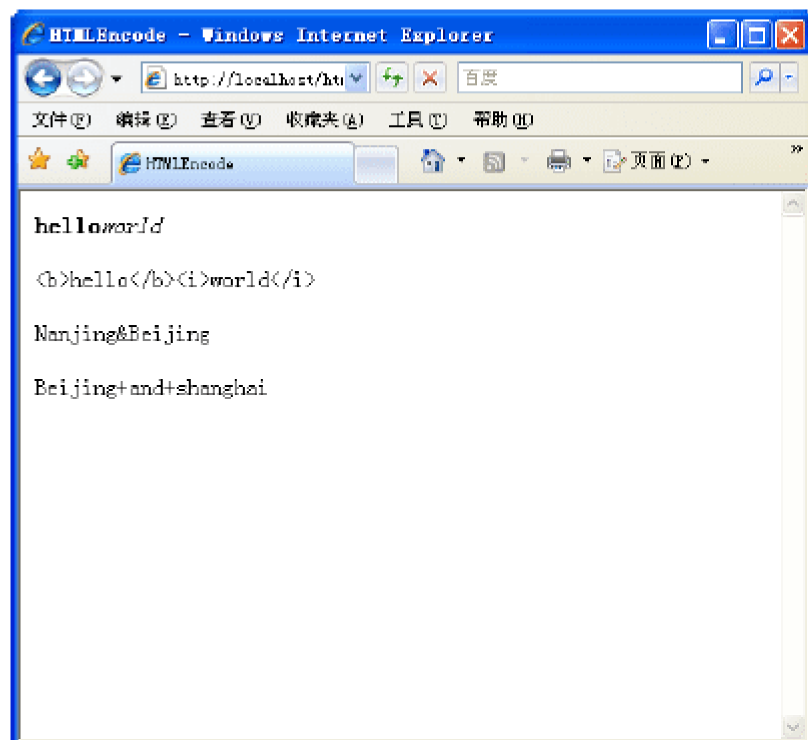


图 5-4　HTML.asp 的效果

5.3.3　URLEncode 方法

URLEncode 方法按 URL 规则对字符串进行编码。当字符串数据以 URL 的形式传递到服务器时，在字符串中不允许出现空格，也不允许出现特殊字符。因此，如果希望在发送字符串之前进行 URL 编码，可以使用 URLEncode 方法。例如下面的代码：

```
<%Response.Redirect ("http://www.baidu.com/baidu?word="&Server.UrlEncode("ASP 学习"))%>
```

此句将把搜索关键字“ASP 学习”添加到 URL 网址的变量值中，通过重定向达到打开相关搜索页面的效果。此例先将“学习”两个汉字更改为 URL 通用的编码%D1%A7%CF%B0，再执行重定向操作就可指向正确的网址：

http://www.baidu.com/baidu?word=ASP%D1%A7%CF%B0

【练习 5-3】编写 ASP 脚本将给出的字符串翻译成可作为 URL 接受的格式。

(1) 创建一个名为 url.asp 的网页，其代码如下所示(如图 5-5 所示):

```
<%
dim urlstring
urlstring="http://www.xxx.com"
response.write"<h2>编码前的字符串如下。"&urlstring&"</h2><br>"
urlstring=Server.URLEncode(urlstring)
response.write"<h2>编码后的字符串如下。"&urlstring&"</h2>"
%>
```

(2) 将 url.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/url.asp 并按 Enter 键，效果如图 5-6 所示。



图 5-5　url.asp 的代码

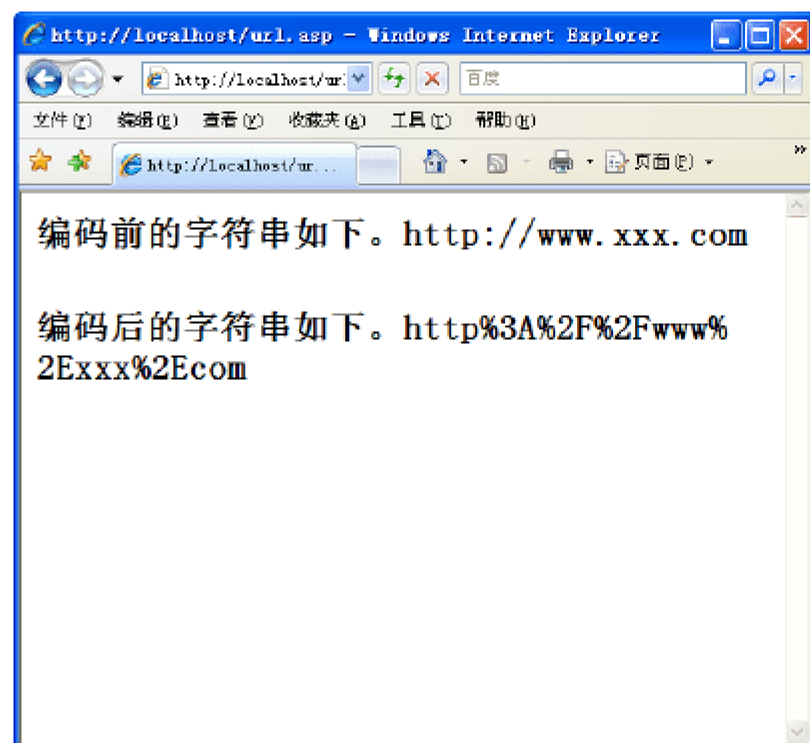


图 5-6　url.asp 的效果

5.3.4　MapPath 方法

MapPath 方法可以将程序指定的相对或虚拟路径映射到服务器上相应的真实路径上，其语法结构如下：

```
Server.MapPath (路径)
```

其中，路径指定要映射到物理目录的相对或虚拟路径。若路径以一个正斜杠(/)或反斜杠(\)开始，则 MapPath 方法返回路径时将路径视为完整的虚拟路径。若路径不是以斜杠开始，则 MapPath 方法返回同 ASP 文件中已有路径的相对路径。

【练习 5-4】利用 Server 对象的 MapPath 方法获取指定文件的绝对路径，程序运行效果如图 5-7 所示。

(1) 创建一个名为 page2.asp 的网页，其代码如下所示(如图 5-8 所示):

```
<%
Response.write("目前网页的虚拟位置：")
```

```
Response.write(request.ServerVariables("PATH_INFO"))
'使用服务器变量 PATH_INFO 映射当前文件的物理路径
Response.write("<BR>")
Response.write("目前网页的绝对位置：")
Response.write(Server.MapPath("page2.asp"))
%>
```

(2) 将 page2.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/page2.asp 并按 Enter 键，效果如图 5-7 所示。

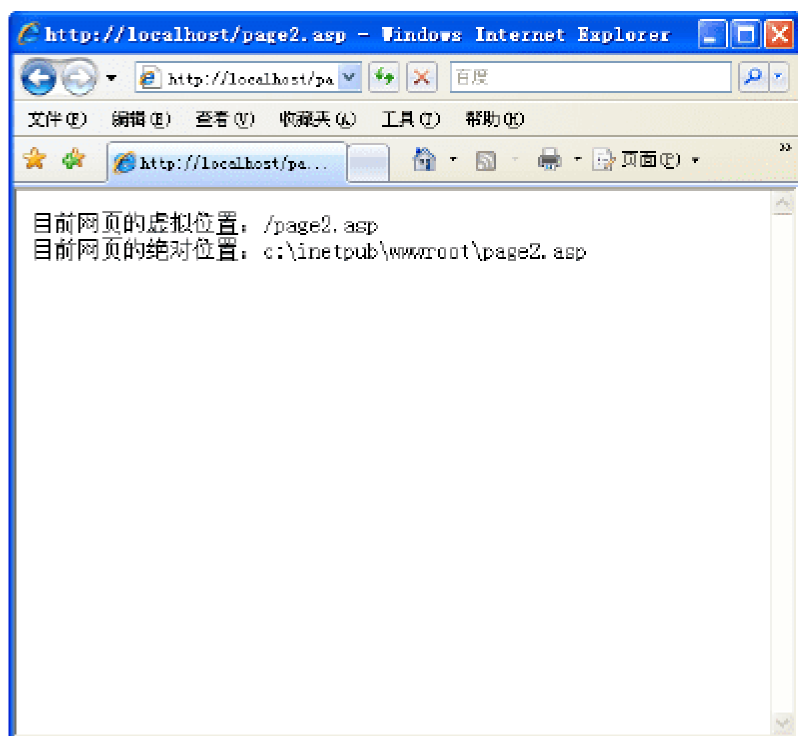


图 5-7　page2.asp 的效果

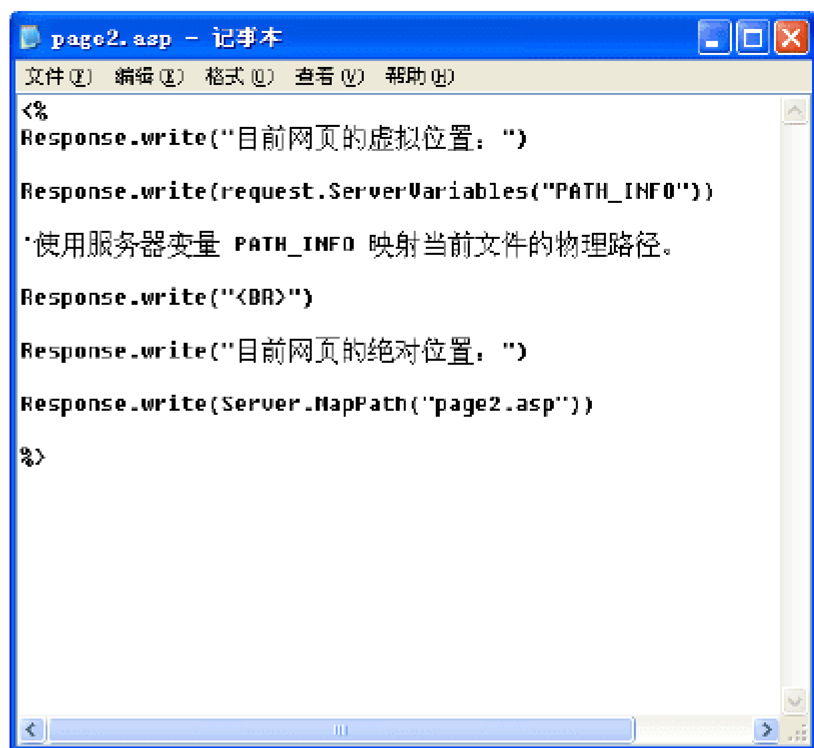


图 5-8　page2.asp 的代码

注意：

MapPath 方法不支持相对路径语法(.)或(..)。例如，使用相对路径../My Webs /MyFile.txt 将返回一个错误。MapPath 方法不检查返回的路径是否正确或在服务器上是否存在。

5.3.5　Transfer 方法

Transfer 方法把执行流程从当前的 ASP 文件转到同一服务器上的另一个 ASP 页面。它的功能和 Response 对象的 Redirect 方法重定向浏览器功能类似，但两者在工作原理上有一定的差别。

使用 Response.Redirect 方法重定向操作的整个过程中，客户端与服务器要进行两次来回的通信。第一次通信是对原始页面的请求，得到一个目标已经改变的应答，第二次通信是请求 Response.Redirect 指向的新页面，得到重定向之后的页面。

使用 Server.Transfer 方法时，客户端与服务器只需进行一次通信，它将终止执行当前的 ASP 页面，执行流程转入另一个 ASP 页面，但新的 ASP 页面仍使用前一个 ASP 页面创建的通信。Transfer 方法需要的网络通信量较小，从而可获得更好的性能和浏览效果，其语法结构如下：

```
Server.Transfer (URL 地址名称)
```

注意：

使用 Server.Transfer 方法实现页面之间的跳转后，浏览器中的 URL 不会改变，因为重定向完全在服务器端进行。

【练习 5-5】使用 Server.Transfer 方法将一个 ASP 文件中的内容传输到另一个 ASP 文件中，其程序运行效果如图 5-9 所示。

(1) 创建一个名为 page3.asp 的网页，其代码如下所示：

```
<p>这是第一个页面！</p>
<%
Response.Write "当前的会话编号为："&Session.SessionID&"<Br>"
Response.Write "下面是执行 Server.Transfer 方法后的结果"&"<Br>"
Server.Transfer("test2.asp")
%>
```

(2) 将 page3.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 创建一个名为 test2.asp 的网页，其代码如下所示(如图 5-10 所示)：

```
<p>这是第二个页面！</p>
<%
Response.Write "当前的会话编号为："&Session.SessionID&"<Br>"
%>
```

(4) 将 test2.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(5) 启动 IE 浏览器后，在地址栏中输入 http://localhost/page3.asp 并按 Enter 键，效果如图 5-9 所示。

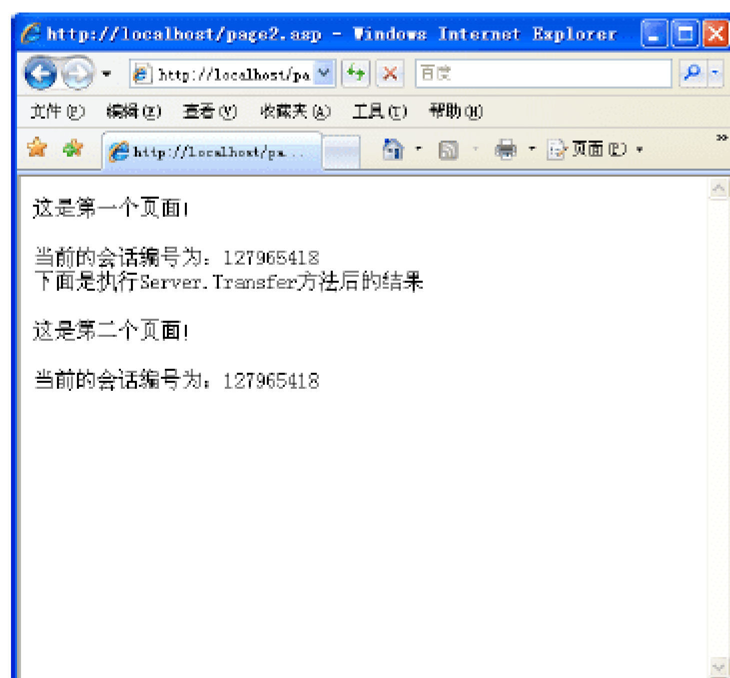


图 5-9　Server.Transfer 的效果

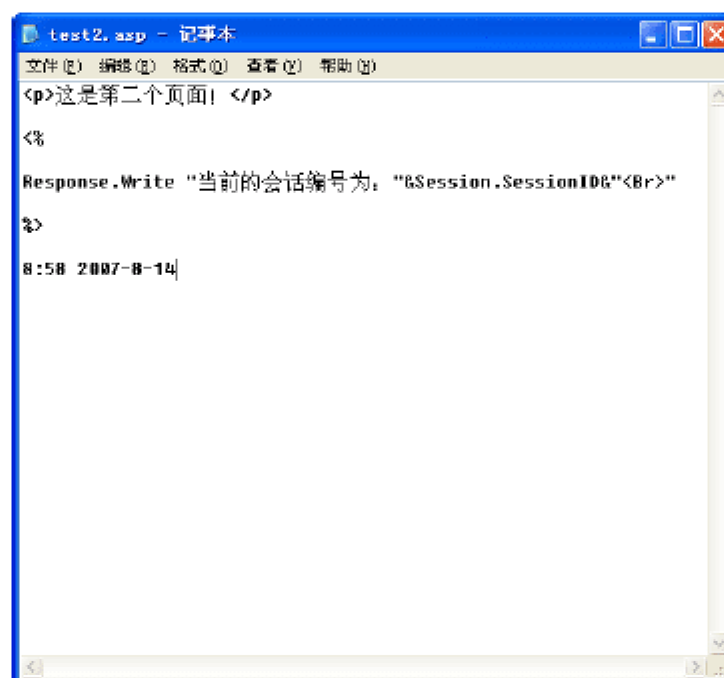


图 5-10　输入第 2 个页面代码

Transfer 方法在默认情况下不会把表单数据或查询字符串从一个页面传递到另一个页面，把该方法的第二个参数设置成 True 后，就可以保留第一个页面的表单数据和查询字符串并加以应用。

注意：

Transfer 方法还可以用来在不同的应用程序的 ASP 文件之间传输，但传输后的文件只能访问作用于初始应用程序的变量，而不访问文件实际上所处的应用程序的变量。

5.3.6　Execute 方法

Execute 方法用来在当前的 ASP 页面执行同一 Web 服务器上指定的另一个 ASP 页面。当指定的 ASP 页面执行完毕，控制流程重新返回原页面发出 Execute 调用的位置。

Execute 方法类似于许多编程语言的过程调用，只不过过程调用是执行一个过程，而 Execute 方法是执行一个完整的 ASP 文件。其语法结构如下：

```
Server.Execute (URL 地址名称)
```

【练习 5-6】使用 Server.Execute 方法，在一个 ASP 文件中调用执行另一个 ASP 文件，效果如图 5-11 所示。

(1) 创建一个名为 page4.asp 的网页，其代码如下所示(如图 5-12 所示)：

```
<p>这是第一个页面！</p>
<%
Response.Write "当前的会话编号为："&Session.SessionID&"<Br>"
Response.Write "下面准备执行 Server.Execute 方法调用第二个页面"&"<Br>"
Server.Execute("test2.asp")
Response.Write "执行完 Server.Execute 方法后返回到第一个页面"&"<Br>"
%>
```

(2) 将 page4.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 创建一个名为 test2.asp 的网页，其代码如下所示：

```
<p>这是第二个页面的内容！</p>
<%
Response.Write "当前的会话编号为："&Session.SessionID&"<Br>"
%>
```

(5) 将 test2.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(6) 启动 IE 浏览器后，在地址栏中输入 http://localhost/page4.asp 并按下 Enter，效果如图 5-11 所示。

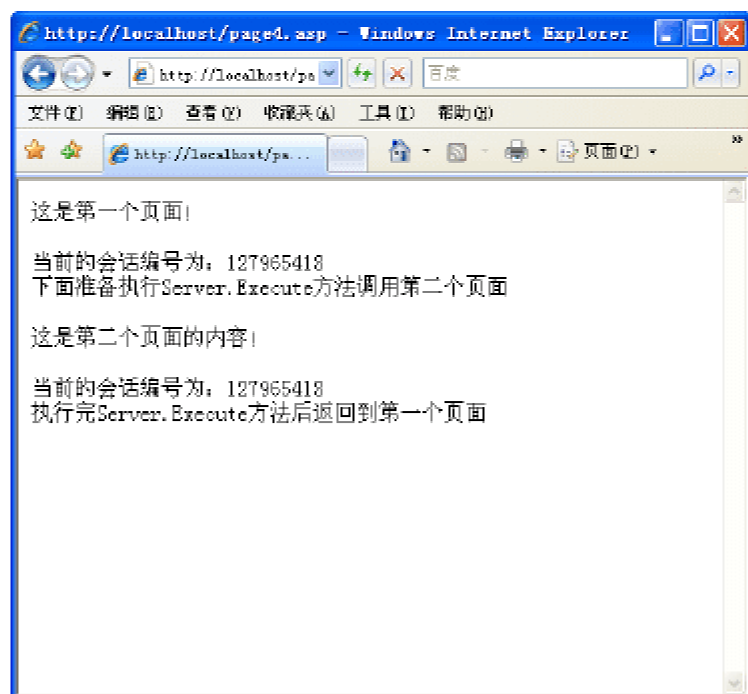


图 5-11　Server.Execute 的效果

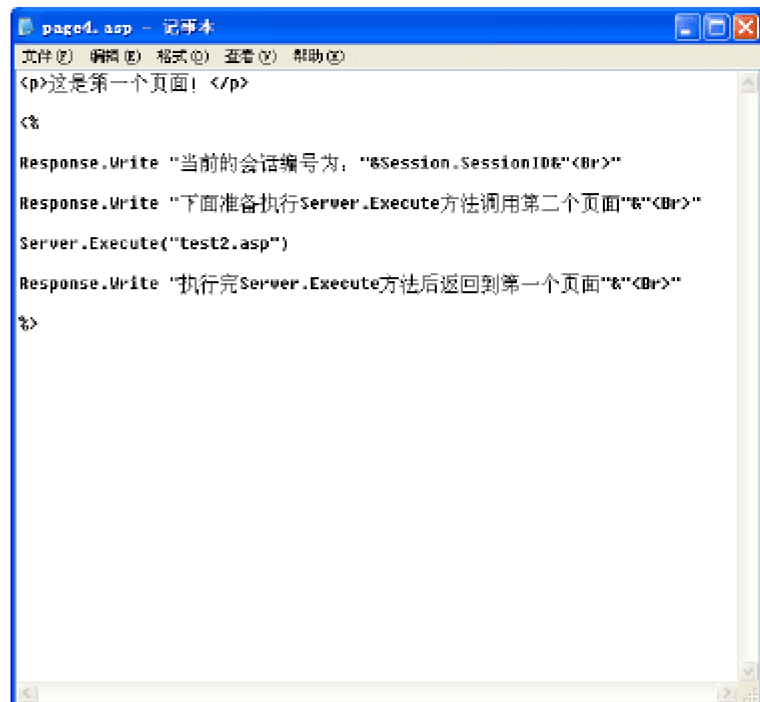


图 5-12　Server.Execute 的代码

注意：

使用 Transfer 方法或 Execute 方法时，最后得到的页面可能不是合法的 HTML 页面，因为最终返回给客户端的页面可能包含多个<HTML>和<BODY>等标记，所以需要多次对页面进行测试。

5.3.7　GetLastError 方法

GetLastError 方法返回一个 ScriptError 对象，用于捕捉当前 ASP 程序的运行错误并向用户返回有用的信息，如错误描述和发生错误的行号等。其语法结构如下：

```
Server.GetLastError ()
```

5.4　习　　题

5.4.1　填空题

(1) Server 对象提供方法__________用以转换 HTML 标记，避免这些特定字符被浏览器进一步解译。

(2) Transfer 方法把执行流程从当前的 ASP 文件转到同一服务器上的另一个 ASP 页面。它的功能和__________的__________重定向浏览器功能类似，但两者在工作原理上有一定的差别。

(3) 使用 Server.Transfer 方法时，客户端与服务器只需进行__________通信，它将终止执行当前的 ASP 页面，执行流程转入另一个 ASP 页面，但新的 ASP 页面仍使用__________创建的通信。

(4) __________返回一个__________，用于捕捉当前 ASP 程序的运行错误并向用户返回有用的信息，如错误描述和发生错误的行号等。

5.4.2　选择题

(1) Server 对象只有一个__________属性，用于指定一个脚本延时的时间期限。

A. TimeOut　　　　B. SessionID

C. ScriptTimeout　　　　D. 以上全是

(2) (　　)方法用来在当前的 ASP 页面执行同一 Web 服务器上指定的另一个 ASP 页面。

A. Abandon　　　　B. MapPath

C. Execute　　D. 以上全是

5.4.3　问答题

(1) 简述 Server 对象的属性。

(2) 简述 Server 对象的方法

5.4.4　操作题

(1) 利用 HTMLEncode 方法将一个指定的字符串按 HTML 的编码输出。

(2) 参考本章练习的操作，使用 Server.Transfer 方法将一个 ASP 文件中的内容传输到另一个 ASP 文件中。

第6章　ASP的常用组件

组件是包含可执行代码的动态链接库(.dll)或可执行文件夹(.exe)，灵活地使用各种组件是建立一个强大的 Web 应用程序的关键。ASP 自带一些基本组件，如 ActiveX Data Objects(ADO)和 Ad Rotator 等，直接使用它们可以高效地完成各种复杂的功能，如数据库访问、广告旋转、文件读取和电子邮件发送等。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 掌握 Ad Rotator 组件的使用方法
- 掌握 Content Rotator 组件的使用方法
- 掌握 Content Linker 组件的使用方法
- 掌握 Browser Capabilities 组件的使用方法
- 掌握 Counters 组件的使用方法

6.1　创建组件实例

ActiveX 组件是一个存在于 Web 服务器上的文件，该文件包含执行某项或一组任务的代码。组件可以执行公用任务，这样网站开发者就不必逐一去创建执行这些任务的代码。例如，使用一个股票行情收报机组件即可以在 Web 页上显示最新的股票报价。

ASP 自带有几个基本组件，网站的开发者还可从第三方开发商购买一些已制作完成的组件，直接运用它们来完成各种各样的任务。此外，还可用任何支持组件对象模型(COM:Component Object Model)的编程语言(如 Visual C++、Java、Visual Basic)来编写自己的组件。组件是可以重复使用的，在 Web 服务器上安装组件后，就可以从 ASP 脚本、ISAPI 应用程序、服务器上的其他组件或另一种 COM 兼容语言编写的程序中调用该组件。

要使用组件提供的对象，首先要创建对象的实例并为这个新的实例分配变量名。使用 ASP 的 Server.CreateObject 方法可以创建对象的实例，然后可使用脚本语言的变量分配指令为对象实例命名。创建对象实例时，必须提供实例的注册名称 PROGID。如下例将创建一个 Ad Rotator 广告条对象：

```
<% Set MyAds = Server.CreateObject("MSWC.AdRotator") %>
```

注意：

必须使用 ASP 的 Server.CreateObject 方法来创建对象实例，否则 ASP 无法跟踪脚本语言中对象的使用。

使用 HTML<OBJECT>标记同样可以创建对象实例，但必须为 RUNAT 属性提供服务器值，同时也要为将在脚本语言中使用的变量名提供 ID 属性组。使用注册名称(PROGID)或注册号码(CLSID)可以识别该对象。下面的例子使用注册名(PROGID)创建 Ad Rotator 对象的实例：

```
< OBJECT RUNAT=Server ID=MyAd PROGID="MSWC.AdRotator"></OBJECT>
```

如表 6-1 所示列出了 ASP 可安装的常用组件。

表 6-1　ASP 可安装的常用组件

组 件 名 称	功　能
Ad Rotator	创建一个 AdRotator 对象，该对象可按指定计划在同一页上自动轮换显示广告
Browser Capabilities	创建一个 BrowserType 对象，该对象决定访问 Web 站点的每个浏览器的性能、类型及版本
Database Access	提供用 ActiveX Data Objects (ADO)对数据库的访问
Content Linking	创建一个 NextLink 对象，该对象可生成 Web 页内容列表，并像书一样将各页顺续连接
File Access 组件	提供文件的输入输出访问
Collaboration Data Objects for NTS 组件	可以快速、简便地在 Web 页上添加收发邮件功能。该组件只适用于 Internet Information Server for Windows NT® Server
MyInfo	创建一个 MyInfo 对象，该对象追踪个人信息，例如站点管理员的姓名、地址及显示选择
Counters	创建一个 Counters 对象，该对象可以创建、保存、增加或检索任意数量的独立计数器
Content Rotator	自动翻转 Web 主页上的 HTML 内容字符串
Page Counter	记录并显示 Web 页被打开的次数

6.2　利用 Ad Rotator 组件制作网站广告条

使用 Ad Rotator 组件可快速在网站上建立一个广告系统，允许在每次访问 ASP 页面时显示新的广告，并且提供了很强的功能，例如：在页面上旋转显示广告图像的能力，跟踪特定广告显示次数的能力以及跟踪客户端在广告上单击次数的能力。

要使用 Ad Rotator 组件的效果，需要用到 3 个文件。AD Rotator 计划文件记录所有广告信息，重定向文件对单击广告条的事件进行处理，广告显示页面则建立和显示广告条。

6.2.1　创建 Ad Rotator 计划文件

Ad Rotator 组件是通过读取 Ad Rotator 计划文件来完成工作的。Ad Rotator 计划文件包括与要显示的图像文件的地点有关的信息以及每个图像的不同属性，下面就是一个标准的 AD Rotator 计划文件：

```
Redirect http://dusiming          '广告被单击后所指向的文件
width 400                         '以像素为单位指定广告的宽度
height 50                         '以像素为单位指定广告的高度
border 0                          '以像素为单位指定广告四周的边框宽度
*                                 '分隔符号
tupwk.gif                         '该广告的图像文件名及位置
http://www.tupwk.com              '单击该广告后要转到的 URL 值
清华文康的广告条                  '图像的替代文字
5                                 '广告的显示频率，频率越高显示的次数也越多
wk040723.gif
http://www.wk.com.cn
wk 的招聘广告
15
Bookhome0722.gif
http://www.Bookhome.com.cn
图书之家
10
```

AD Rotator 计划文件由两部分组成。第一部分设置应用于轮换安排中所有广告图像的参数；第二部分指定每个单独广告的文件和位置信息以及应当接收的每个广告的显示时间所占百分比。这两部分由全是星号(*)的一行隔开。

在第一部分中有 4 个全局参数，每个参数都由一个关键字和值组成。Redirect 行指出该广告链接到的 URL 地址，星号上面的其余 3 行简单说明如何显示广告。Width 行和 Height 行以像素为单位指定网页上广告的宽度和高度，默认值是 440 和 60 像素。Border 行以像素为单位指定广告四周超链接的边框宽度，默认值为 1 像素，如果将该参数设置为 0，该广告则将没有边框。这 4 个参数都是可选的，如果用户未指定它们的值，则 Ad Rotator 组件将使用默认的值。

星号下面的第二部分以每 4 行为一个单位描述每个广告的具体内容。每个广告的描述包含图像文件的 URL、广告的主页 URL(连字符“-”可指出该广告没有链接)和图像的替代文字以及指定该页与其他页交替显示频率的数值。要确定广告显示的频率，可以将计划文件中所有广告的权值相加，在该例中总数是 30，那么 hsbc 的广告权值为 5，这意味着每调用 6 次 Ad Rotator 组件，则显示一次广告。

6.2.2　设置广告图像重定向文件

广告条放置到网站后，用户对广告条进行单击操作后，ASP 就会打开重定向文件。重定向文件通常是用户创建的文件，包含用来解析由 Ad Rotator 对象发送的查询字符串的脚本，并将用户重定向到与用户所单击的广告所相关的 URL。用户也可以将脚本包含到重定向文件中，以便统计单击某一特定广告的用户数目，并将这一信息保存到服务器上的某一文件中，如下例所示：

```
<%
'将单击情况记录到 Web 服务器日志文件中
Response.AppendToLog Request.QueryString("url")
'重定向到广告指定的站点
Response.Redirect (Request.Querystring("url"))
%>
```

注意：

Redirect 行所指示的不是为广告本身指定的 URL，而是将调用的中间页面的 URL。这样，就可以通过这个中间页面跟踪单击广告的次数。该 Redirect URL 将与包含两个参数的查询字符串一起调用特定广告主页的 URL 和图像文件的 URL。

6.2.3　创建网站广告显示页面

如果 ASP 在页面中调用 Ad Rotator 组件，首先必须使用 Server.CreateObject 方法实例化 AdRotator 对象。Ad Rotator 组件的 PROGID 属性是 MSWC.AdRotator，完整的代码如下：

```
< % Set ad = Server.CreateObject("MSWC.AdRotator") %>
<%= ad.GetAdvertisement("/ads/adrot.txt") %>
```

Ad Rotator 组件支持的唯一方法是 GetAdvertisement，它只有一个参数 AdRotator 是计划文件的名称。注意指向文件的路径是当前虚拟目录的相对路径，物理路径是不允许的。GetAdvertisement 方法从 Rotator 计划文件中获取下一个计划广告的详细说明并将其格式化为 HTML 格式。下面的 HTML 由 GetAdvertisement 方法生成且被添加到网页的输出中，以便显示 Rotator 计划文件中的下一个广告。

```
<A HREF="http://zhangshihua?url=http://www.ut.com.cn&image=bookhome0722.gif" >
<IMG SRC="bookhome0722.gif" ALT="图书之家" WIDTH=400 HEIGHT=50 BORDER=0></A>
```

使用 Ad Rotator 组件还可以直接通过对象属性，而不是计划文件中的设置来直接控制某些广告特性，其可用属性如下。

- Border 属性：指定广告边框的大小。

- Clickable 属性：指定广告是否有与之相关联的超级链接。默认值为 True，表示有相关联的超级链接。
- TargetFrame 属性：指定要打开与广告相关联的超级链接的框架名称。

【练习 6-1】创建一个显示广告条的页面。

(1) 将如图 6-1 所示的 3 幅 gif 图片文件复制至 C:\Inetpub\wwwroot 文件夹中。



1.gif　　2.gif　　3.gif

图 6-1　图片效果

(2) 创建一个名为 adrot.txt 的文件，其代码如下所示：

```
Redirect adredir.asp
width 100
height 50
border 0
*
1.gif
http://www.baidu.com
百度搜索
5
2.gif
http://www.sina.com
新浪网站
15
3.gif
http://www.sohu.com
搜狐网站
10
```

(3) 将 adrot.txt 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(4) 创建一个名为 adrot.asp 的文件，其代码如下所示：

```
<HTML>
<body>
<%set myad=Server.CreateObject("MSWC.adrotator")
          myad.Border = 0
          myad.Clickable = true
          myad.TargetFrame = AdFrame
%>
<%=myad.getadvertisement("/adrot.txt")%>
```

```
<HR SIZE="1" COLOR="#000000">
<Center><A HREF="adrot.asp">重新刷新本页</A></center>
    </BODY>
</HTML>
```

(5) 将 adrot.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(6) 启动 IE 浏览器后，在地址栏中输入 http://localhost/adrot.asp 并按 Enter 键，网页效果将如图 6-2 所示(用户每刷新一次网页，浏览器将显示不同的网站标志)。

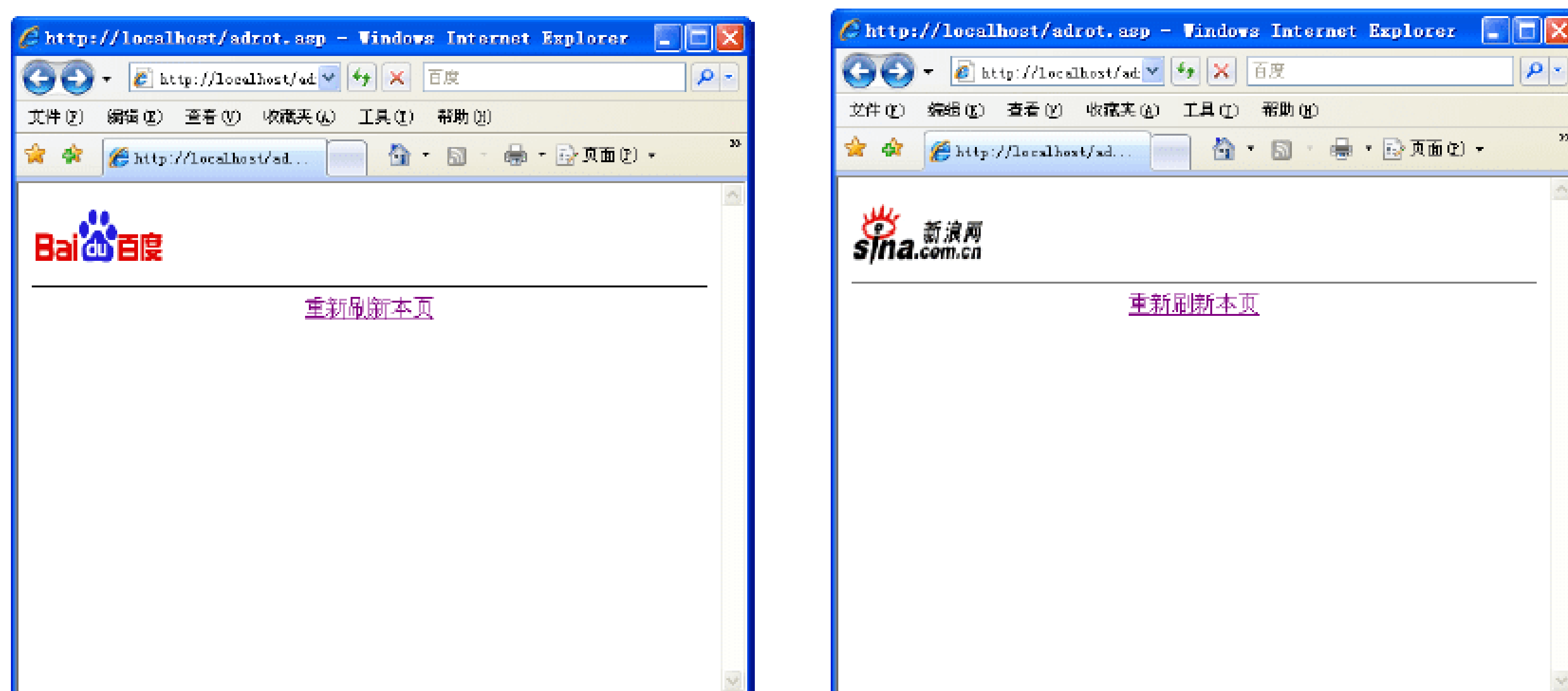


图 6-2　adrot.asp 的效果

(7) 重新创建 adrot.asp 文件，其代码如下所示：

```
<%
'将单击情况记录到 Web 服务器日志文件中
Response.AppendToLog Request.QueryString("url")
'重定向到广告指定的站点
Response.Redirect (Request.Querystring("url"))
%>
```

(8) 将上面创建的 adrot.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。这时，如果用户运行该文件，并再次单击图 6-2 所示网页中的网站广告图片，浏览器将会自动跳转到相关的网站，如图 6-3 所示。

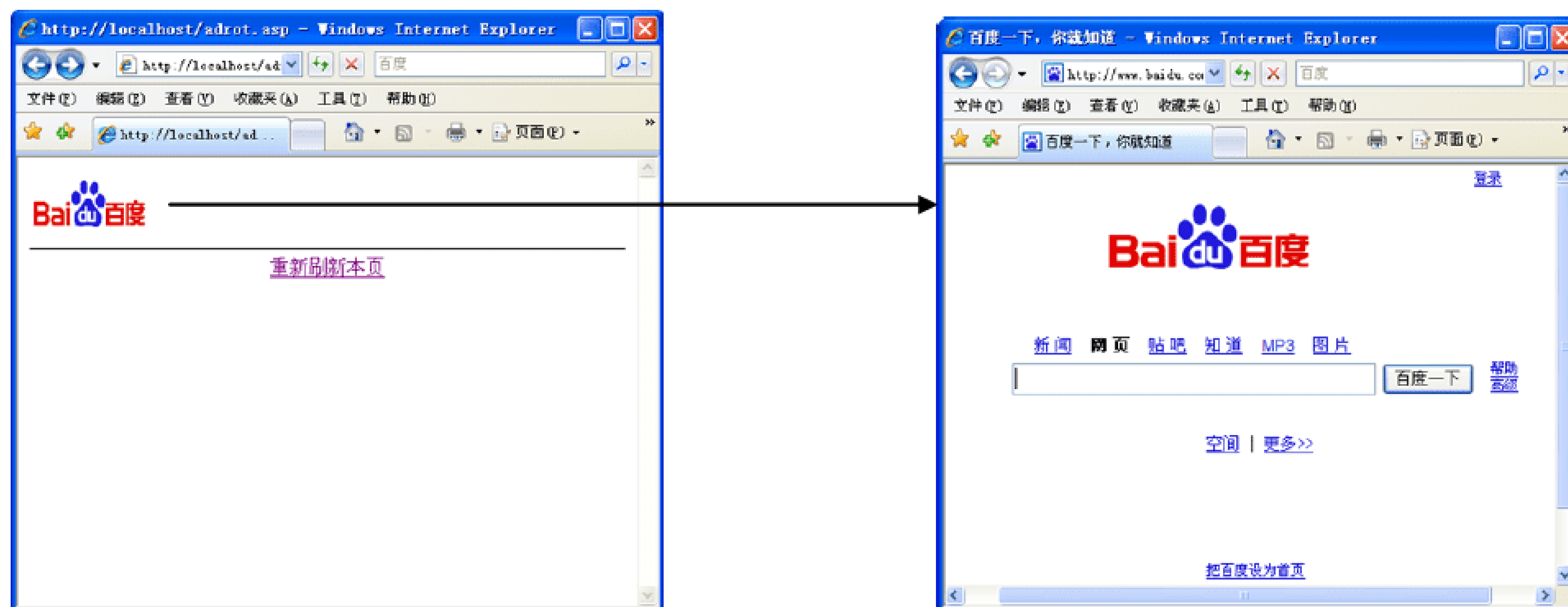


图 6-3　重新创建后的效果

6.3　利用 Content Rotator 组件自动切换内容

Content Rotator 组件通过读取计划文件来完成网页内容的显示，通常是自动轮换显示一些 HTML 内容。当用户请求 Web 页时，Content Rotator 组件从内容计划文件中取得待显示的内容。待显示内容可以是 HTML 能够表达的任何内容，包括文本、图像和超链接等。

内容计划文件是一个文本文件，提供要使用的各个文本字符串的列表，每个要显示的列表部分以两个百分号开始，格式如下：

```
%% [# 权重] [//注释]
待轮换显示的内容
```

权重设置了每个条目出现在返回页面中的频率列表的显示频率，取值范围为 0~65 535，默认值为 1。

Content Rotator 组件有 ChooseContent 和 GetAllContent 两个方法。ChooseContent 方法用于从内容计划文件中得到一项显示内容，而 GetAllContent 方法用于显示内容计划文件中的所有内容。

【练习 6-2】创建一个从可选列表中随机显示内容的页面，程序运行效果如图 6-4 所示。

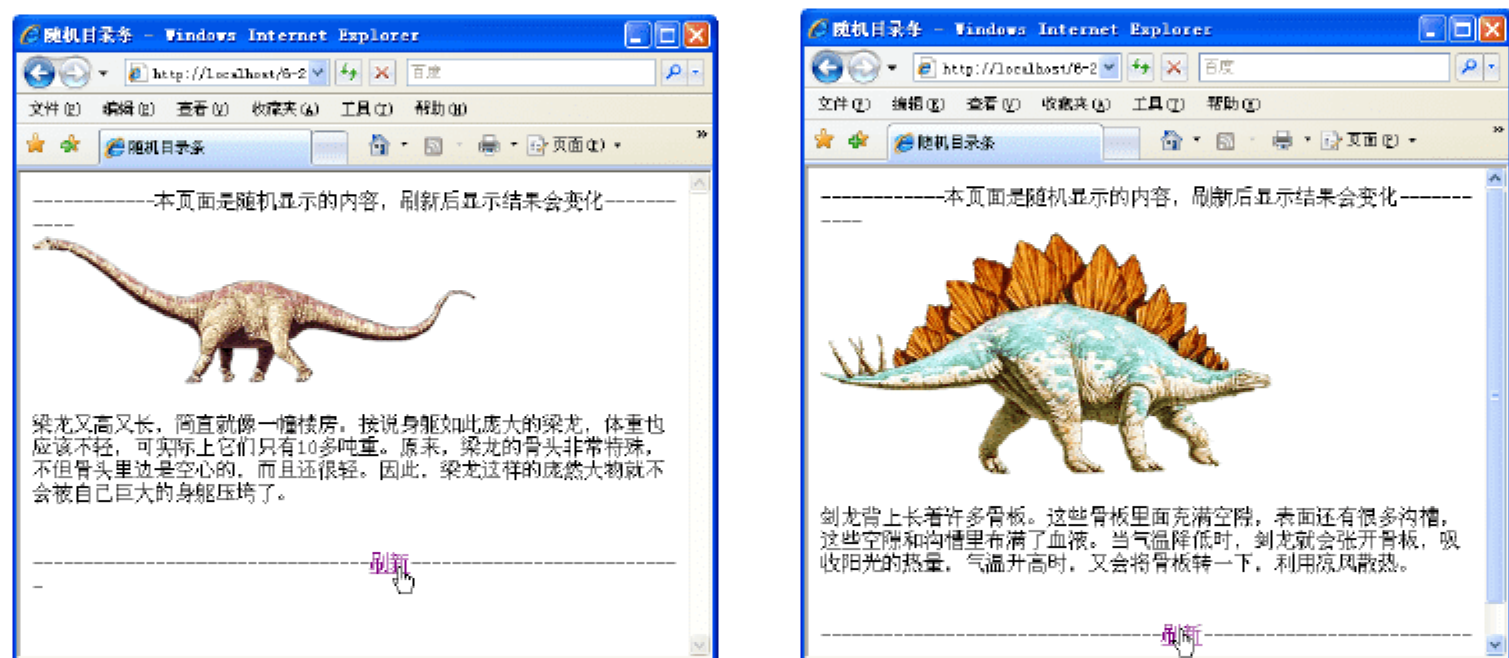


图 6-4　网页效果

(1) 将如图 6-5 所示的图片文件 t1.asp、t2.asp、t3.asp 和 t4.asp 复制至 C:\Inetpub\wwwroot 文件夹中。



图 6-5　图片效果

(2) 创建一个名为“6-2-内容随机显示.asp”的文件，其代码如下所示：

```
%% #1
```

```
' 显示阿尔伯特龙
<A Href="6-2-1.asp"><Img Src="t1.gif" Border=0></A>
<p>阿尔伯特龙的身躯要小一些，但它们却更令其他动物害怕。因为它们奔跑的速度极快，据估计，短距离内可达时速 30 多公里。阿尔伯特龙的可怕之处还在于它的嘴巴特别大，里边排满了尖利的牙齿，能咬穿坚硬的骨头，更不用说其他恐龙的厚皮了。</P>
%% #2//显示梁龙
%%//再加一行注释
%%//第三行注释
<A Href="6-2-2.asp"><Img Src="t2.gif" Border=0></A>
<p>梁龙又高又长，简直就像一幢楼房。按说身躯如此庞大的梁龙，体重也应该不轻，可实际上它们只有 10 多吨重。原来，梁龙的骨头非常特殊，不但骨头里边是空心的，而且还很轻。因此，梁龙这样的庞然大物就不会被自己巨大的身躯压垮了。</P>
%% #4//显示慈母龙
<A Href="6-2-3.asp"><Img Src="t3.gif" Border=0></A>
<p>慈母龙每次能生 25 个蛋，这 25 只小恐龙每天要吃掉几百斤鲜嫩的植物，慈母龙需要不辞劳苦地到处寻找食物。如果真是这样的话，它们是无愧于慈母龙这个称号的。</P>
%% #5//显示剑龙
<A Href="6-2-2.asp"><Img Src="t4.gif" Border=0></A>
<p>剑龙背上长着许多骨板。这些骨板里面充满空隙，表面还有很多沟槽，这些空隙和沟槽里布满了血液。当气温降低时，剑龙就会张开骨板，吸收阳光的热量，气温升高时，又会将骨板转一下，利用凉风散热。</P>
```

(3) 选择“文件”|“另存为”命令，将上面输入的代码以文件名“6-2-内容随机显示”保存至 C:\Inetpub\wwwroot 文件夹中。

(4) 在记事本工具中重新输入以下代码:

```
<Html>
<Head>
<Title>
随机目录条
</Title>
</Head>
<Body>
------------本页面是随机显示的内容，刷新后显示结果会变化-----------<br>
<%
set NextTip=Server.CreateObject("MSWC.ContentRotator")
%>
<%=NextTip.ChooseContent("6-2-内容随机显示.TXT")%>
<br>
--------------------------------<A Href="6-2-内容随机显示.asp">刷新</A>--------------------------
</Body>
</Html>
```

(5) 将“6-2-内容随机显示.asp”文件保存至 C:\Inetpub\wwwroot 文件夹中。

(6) 启动 IE 浏览器后，在地址栏中输入 http://localhost/6-2-内容随机显示.asp 并按 Enter 键，网页效果将如 6-4 所示。

注意：

如果将“6-2-内容随机显示.asp”程序脚本修改为如下所示，运行 ASP 网页后会发现 context.txt 所有内容被解释执行，并且分类之间会自动添加<hr>水平线，如图 6-6 所示。

```
<Html>
<Head>
<Title>
随机目录条
</Title>
</Head>
<Body>
------------本页面是随机显示的内容，刷新后显示结果会变化-----------<br>
<%
set NextTip=Server.CreateObject("MSWC.ContentRotator")
%>
<%=NextTip.GetAllContent("6-2-内容随机显示.TXT")%>
<br>
---------------------------------<A Href="6-2-内容随机显示.asp">刷新</A>---------------------------
</Body>
</Html>
```



图 6-6　网页效果

6.4　使用 Content Linker 组件建立页面索引

Content Linker 组件可在一系列相互关联的页面中建立一个目录表，在它们中间建立动态链接，并自动生成和更新目录表及先前和后续的 Web 页的导航链接。该组件常用于需要建立大量页面为访问者提供导航，例如联机报刊、电子读物网站以及论坛邮件等。

6.4.1　使用 Content Linker 组件创建列表文件

要使用 Content Linker 组件，必须首先创建一个列表文件。Content Linker 组件正是通过读取这个文件来获得处理链接的所有页面的信息。列表文件都是纯文本格式的，如下例所示：

```
Xajh01.asp	第一章　灭门
Xajh02.asp	第二章　聆秘
Xajh03.asp	第三章　救难
……
Xajh40.asp	第四十章　曲谐
Xajh41.asp	后记
```

在列表文件中，第一列 URL 是与页面相关的超链接地址，第二列是在网页上显示的链接描述，并且这两列必须用 Tab 键来隔开。

注意：

列表文件中的 URL 地址是虚拟或相对的，只能引用本地服务器上的文件。如果以 http://、//或\\开头，则不被支持，也不被处理。

6.4.2　使用 Content Linker 组件的方法

下面列出了 Content Linker 组件所有可使用的方法，如表 6-2 所示：

表 6-2　Content Linker 组件的方法

组 件 方 法	说　　明
GetListCount(目标文件名)	显示组件中包含的链接的文件数目
GetListIndex(目标文件名)	显示当前页在这些链接文件中的前后位置索引值
GetNextDescription(目标文件名)	显示链接文件中下一个文件的描述
GetNextURL(目标文件名)	显示链接文件中下一个文件的 URL 地址
GetNthDescription(目标文件名, N)	显示链接文件中第 N 个网页的描述
GetNthURL(目标文件名, N)	显示链接文件中第 N 个网页的文件 URL 地址
GetPreviousDescription(目标文件名)	显示链接文件中前一个文件的描述
GetPreviousURL(目标文件名)	显示链接文件中前一个文件的 URL 地址

【练习 6-3】使用 Content Linker 组件从列表文件中创建一个图书目录表，程序运行效果如图 6-7 所示。

(1) 创建一个名为 list.txt 的文件，其代码如下所示：

```
Xajh01.asp 第一章 灭门
Xajh02.asp 第二章 聆秘
Xajh03.asp 第三章 救难
Xajh04.asp 第四章 坐斗
Xajh05.asp 第五章 治伤
Xajh06.asp 第六章 洗手
Xajh07.asp 第七章 授谱
Xajh08.asp 第八章 面壁
Xajh09.asp 第九章 邀客
Xajh10.asp 第十章 传剑
Xajh11.asp 第十一章 聚气
Xajh12.asp 第十二章 围攻
Xajh13.asp 第十三章 学琴
Xajh14.asp 第十四章 论杯
Xajh15.asp 第十五章 灌药
……………………………
Xajh30.asp 第三十章   密议
Xajh31.asp 第三十一章 绣花
Xajh32.asp 第三十二章 并派
Xajh33.asp 第三十三章 比剑
Xajh34.asp 第三十四章 夺帅
Xajh35.asp 第三十五章 复仇
Xajh36.asp 第三十六章 伤逝
Xajh37.asp 第三十七章 迫娶
Xajh38.asp 第三十八章 聚歼
Xajh39.asp 第三十九章 拒盟
Xajh40.asp 第四十章   曲谐
Xajh41.asp    后记
```

(2) 将 list.txt 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 创建一个名称 list.asp 的文件，其代码如下所示(如图 6-8 所示):

```
<Html>
<Title>金庸小说《笑傲江湖》图书目录</Title>
<Center><H1>笑傲江湖</H1></Center>
<Hr><Ul>
<%Set ML = Server.CreateObject("MSWC.Nextlink")%>
<%
intCount = ML.GetListCount("list.TXT")
For i = 1 To intCount
%>
<Li>   
<a href="<%=ML.GetNthURL("list.TXT", i) %>">
```

```
<%=ML.GetNthDescription("list.TXT", i) %>
</a>
<Br>
<%Next%>
</Ul>
```

(4) 将 list.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(5) 启动 IE 浏览器后，在地址栏中输入 http://localhost/list.asp 并按 Enter 键，网页效果将如 6-7 所示。

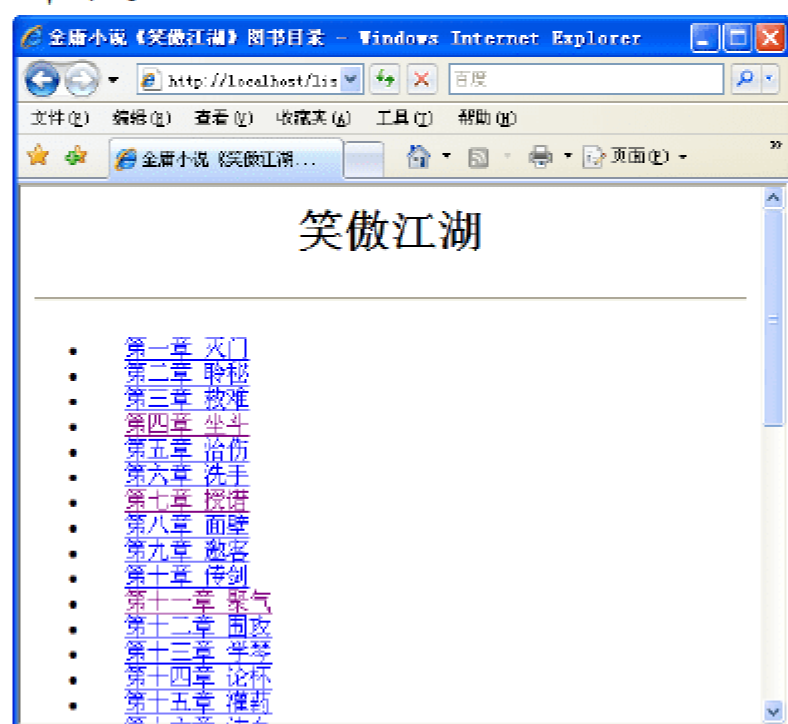


图 6-7　list.asp 的效果

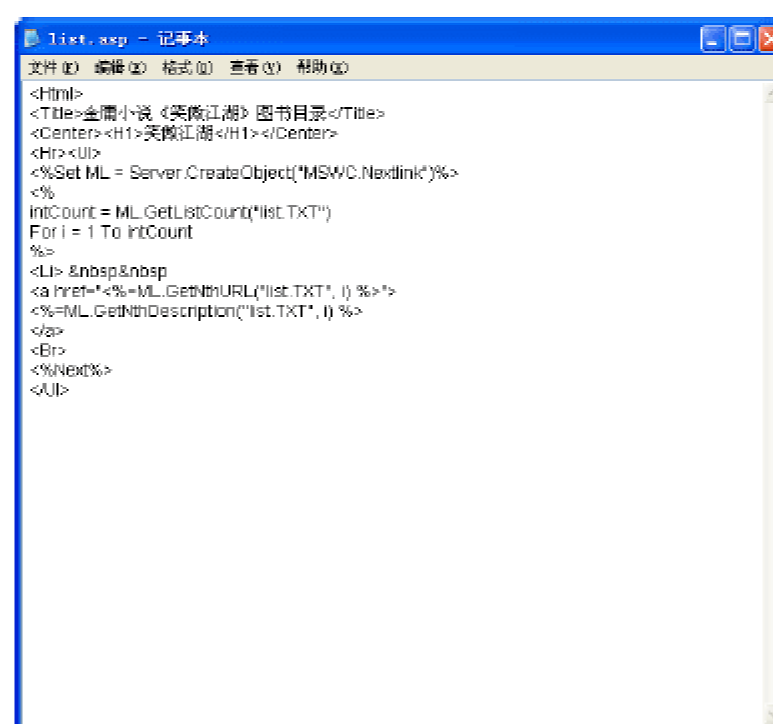


图 6-8　list.asp 的代码

(6) 将如图 6-9 所示的网页文件(Xajh01.asp ~ Xajh41.asp)复制至 C:\Inetpub\wwwroot 文件夹中。

(7) 在浏览器中单击 list.asp 页面中对应的目录名即可打开相应的内容页面，如图 6-10 所示。

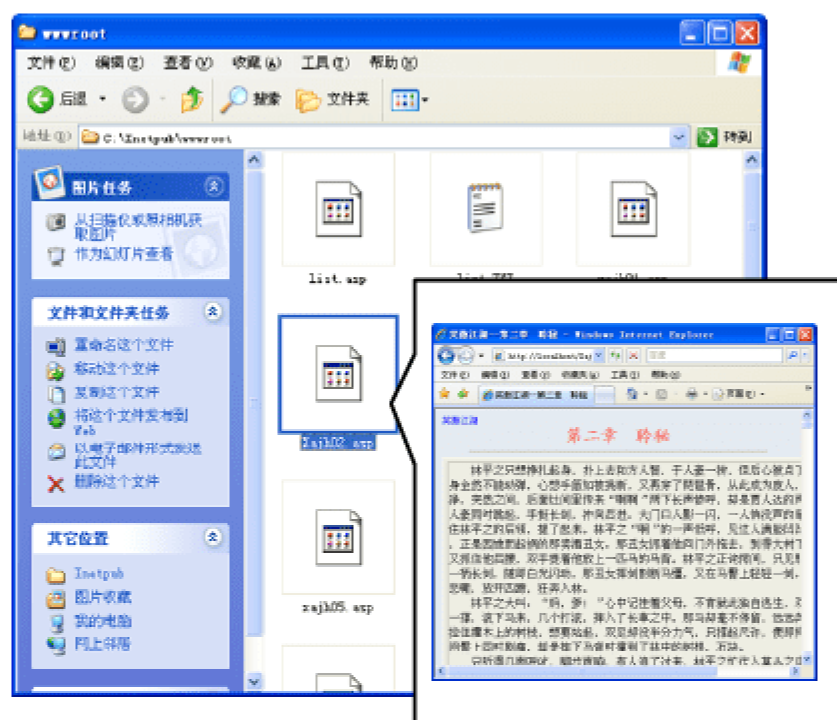


图 6-9　复制网页文件

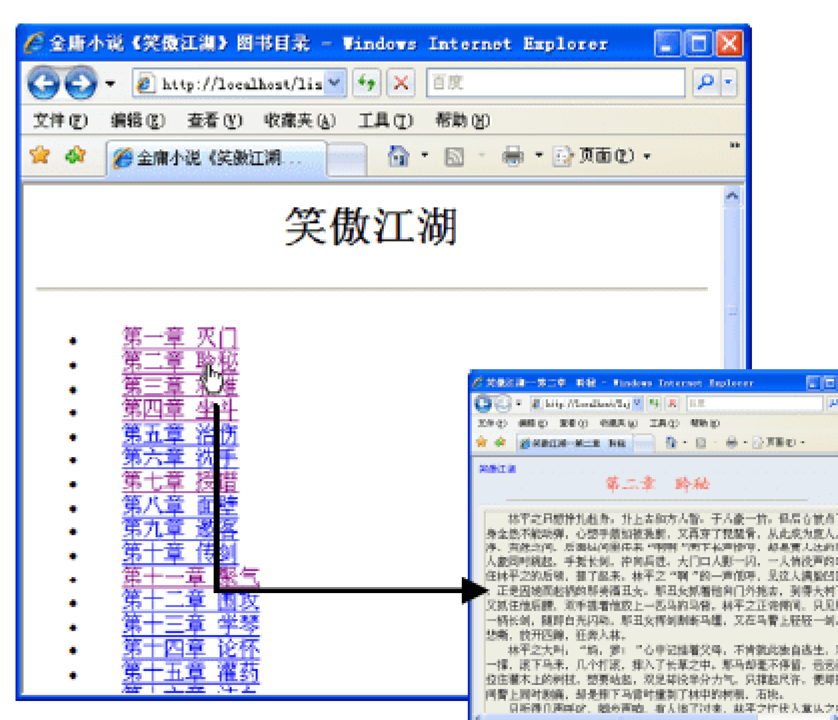


图 6-10　打开网页

6.5　使用 Browser Capabilities 组件建立页面索引

Browser Capabilities 组件用于检测客户端浏览器的能力。通过该组件，可以知道浏览

器的名称、版本，以及是否支持框架、ActiveX 控件、Cookie 和脚本程序等。

Browser Capabilities 组件的工作原理如下：当客户端浏览器向服务器发送页面请求时会自动发送一个 User Agent HTTP 标题，而该标题是一个声明浏览器及其版本的 ASCII 字符串。而 Browser Capabilities 组件就将 User Agent 映射到文件 Browscap.ini 中所注明的浏览器，并通过 BrowserType 对象的属性来识别客户浏览器。

若该对象在 browscap.ini 文件中找不到与该标题匹配的项，那么将使用默认的浏览器属性。若该对象既未找到匹配项且 browscap.ini 文件中也未指定默认的浏览器设置，则它将每个属性都设为字符串 UNKNOWN。

在默认情况下，browscap.ini 文件被存放在 C:\WINDOWS\system32\inetsrv 目录中，用户可以编辑这个文本文件，以添加属性或根据最新发布的浏览器版本的更新文件来修改该文件。

Browser Type 对象包含的属性如表 6-3 所示。

表 6-3　Browser Type 对象的属性

属　　性	说　　明
Browser	指定浏览器的名称
Version	指定浏览器的版本号
Majorver	指定主版本号
Minorver	指定副版本号
Frames	指定浏览器是否支持框架
Tables	指定浏览器是否支持表格
Cookies	指定浏览器是否支持 Cookie
Backgroundsounds	指定浏览器是否支持背景音乐
VBScript	指定浏览器是否支持 VBScript
JavaScript	指定浏览器是否支持 JScript
JavaApplets	指定浏览器是否支持 Java 小程序
ActiveXControls	指定浏览器是否支持 ActiveX 控件
Beta	指定浏览器是否是测试版
Cdf	指定浏览器是否支持频道定义文件

【练习 6-4】使用 Browser Capabilities 组件检测浏览器的常用属性，程序运行效果如图 6-11 所示。

(1) 创建一个名为 Browser.asp 的文件，其代码如下所示(如图 6-12 所示):

```
<Center><H1>您的浏览器性能如下：</H1></Center>
<Hr>
<%set bc=server.CreateObject("mswc.browsertype")%>
<%if bc.frames=true then%>浏览器支持多窗口(frames)显示
```

```
<%else%>浏览器不支持多窗口(frames)显示
<%end if%><br>
<%if bc.backgroundsounds=true then%>浏览器可以播放背景音乐(backgroundsounds)
<%else%>浏览器不能播放背景音乐(backgroundsounds)
<%end if%><br>
<%if bc.tables=true then%>浏览器支持表格(tables)显示
<%else%>浏览器不支持表格(tables)显示
<%end if%><br>
<%if bc.beta=true then%>您的浏览器是一测试版(beta)
<%else%>你的浏览器是一正式版
<%end if%><br>
<%if bc.activexcotrols=true then %>浏览器支持 active 控制
<%else%>浏览器不支持 active 控制
<%end if%><br>
<%if bc.cookies=true then%>浏览器支持 cookie 功能
<%else%>浏览器不支持 cookie 功能
<%end if%><br>
<%if bc.vbscript=true then%>浏览器支持 vbscript
<%else%>浏览器不支持 vbscript
<%end if%><br>
<%if bc.jscript=true then%>浏览器支持 jscript
<%else%>浏览器不支持 jscript
<%end if%>
```

(2) 将 Browser.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Browser.asp 并按 Enter 键，网页效果将如 6-11 所示。

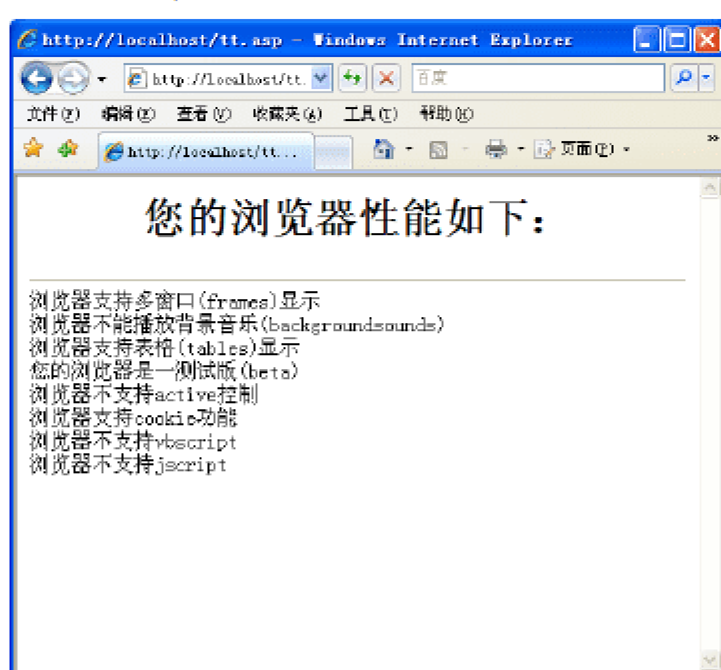
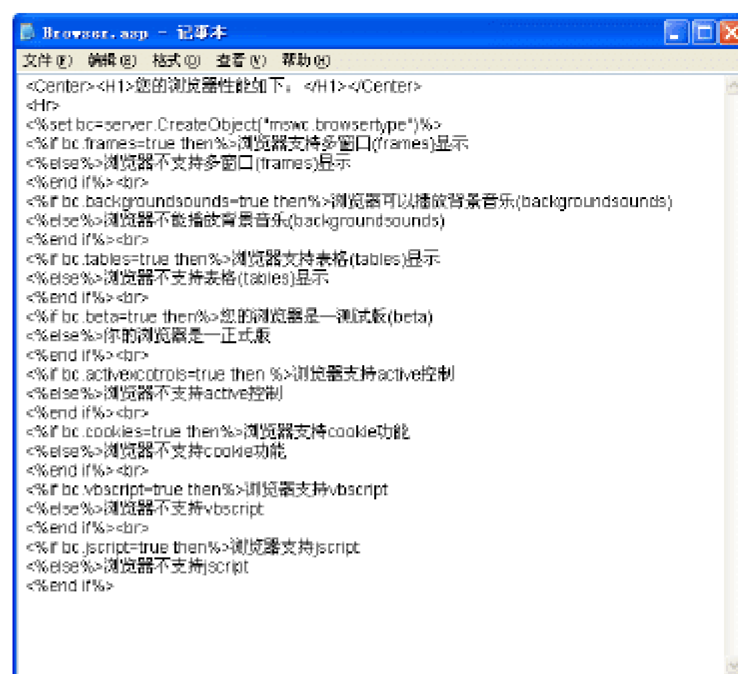


图 6-11　Browser.asp 的效果



```
<Center><H1>您的浏览器性能如下：</H1></Center>
<Hr>
<%set bc=server.CreateObject("mswc.browsertype")%>
<%if bc.frames=true then%>浏览器支持多窗口(frames)显示
<%else%>浏览器不支持多窗口(frames)显示
<%end if%><br>
<%if bc.backgroundsounds=true then%>浏览器可以播放背景音乐(backgroundsounds)
<%else%>浏览器不能播放背景音乐(backgroundsounds)
<%end if%><br>
<%if bc.tables=true then%>浏览器支持表格(tables)显示
<%else%>浏览器不支持表格(tables)显示
<%end if%><br>
<%if bc.beta=true then%>您的浏览器是一测试版(beta)
<%else%>你的浏览器是一正式版
<%end if%><br>
<%if bc.activexcotrols=true then %>浏览器支持active控制
<%else%>浏览器不支持active控制
<%end if%><br>
<%if bc.cookies=true then%>浏览器支持cookie功能
<%else%>浏览器不支持cookie功能
<%end if%><br>
<%if bc.vbscript=true then%>浏览器支持vbscript
<%else%>浏览器不支持vbscript
<%end if%><br>
<%if bc.jscript=true then%>浏览器支持jscript
<%else%>浏览器不支持jscript
<%end if%>
```

图 6-12　Browser.asp 的代码

6.6　使用 Counters 组件制作计数器

Counters 组件用于创建 Counters 对象，该对象可创建一个或多个计数器，这些计数器

用于跟踪某一网页或某一网站访问次数的信息。一个计数器包含一个整数的持久值，一旦创建了计数器，它将一直持续下去直到被删除为止。

Counters 对象有 4 个方法，如表 6-4 所示。

表 6-4　Counters 对象的方法

方　法	说　明
Get	用于返回计数器的当前值，如果计数器不存在，则创建一个计数器并将值设为 0
Increment	使计数器的值递增 1，如果计数器不存在，则创建一个计数器并将值设为 1
Set	将计数器的值设为指定的值，如果计数器不存在，则创建一个计数器并将值设为指定值
Remove	从 Counter 对象中删除一个计数器

一个站点只能创建一个计数器组件，但该组件可以有多于一个的计数器。Counters 对象通常是在 Global.asa 文件中用以下命令创建的：

```
<OBJECT RUNAT=Server SCOPE=Application ID=Counter PROGID="MSWC.Counters">
</OBJECT>
```

注意：

计数器不受作用域限制。一旦创建了计数器，那么站点上的任何页都可以检索和控制它的值。

【练习 6-5】使用 Counters 组件创建一个投票计数器，程序运行效果如图 6-13 所示。

(1) 创建一个名为 remove.asp 的文件，其代码如下所示：

```
<%
Set Counters=Server.Createobject("MSWC.Counters")
  vote = Request.QueryString("site")
  clear = Request.QueryString("clear")
  Counters.Increment(vote)
  If Not clear="" Then
    Counters.Remove("动作片")
    Counters.Remove("爱情片")
    Counters.Remove("伦理片")
  End If
%>
您最喜欢哪种类型的电影，请投票:
<Hr>
<FORM NAME="filmtype" METHOD="GET" ACTION="remove.asp">
<Input type="RADIO" NAME="site" VALUE="动作片">动作片
<Input type="RADIO" NAME="site" VALUE="爱情片">爱情片
<Input type="RADIO" NAME="site" VALUE="伦理片">伦理片
<Br><Br><INPUT TYPE="SUBMIT" VALUE="我要进行投票">
```

```
</FORM>
<Hr>
当前的投票结果:<BR>
动作片: <% =Counters.Get("动作片") %><BR>
爱情片: <% =Counters.Get("爱情片") %><BR>
伦理片: <% =Counters.Get("伦理片") %>
<FORM NAME="Clear Counters" METHOD="GET" ACTION="remove.asp">
<INPUT TYPE="SUBMIT" VALUE="重新计票" NAME="clear">
</FORM>
```

(2) 将 remove.asp 保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在浏览器地址栏中输入 http://localhost/remove.asp 并按 Enter 键，网页效果将如图 6-13 所示。

(4) 在 remove.asp 页面中选中投票的单选按钮后，单击“我要进行投票”按钮，页面将自动将投票结果显示在网页的“当前投票结果”选项区域中(如图 6-14 所示)。单击页面中的“重新计票”按钮，ASP 网页将自动清空“当前投票结果”选项区域中的投票结果。



图 6-13　remove.asp 的效果

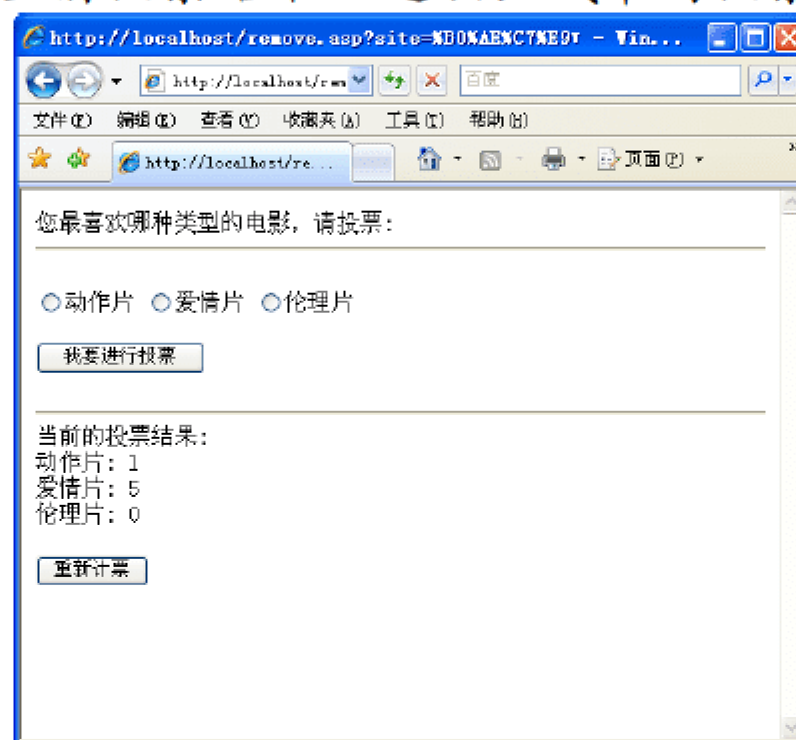


图 6-14　显示投票结果

6.7　使用 Page Counter 组件

Page Counter 组件用于创建 PageCounter 对象，该对象用来记录和显示 Web 页被打开的次数。每隔一定的时间，此对象将当前的页面访问次数写入一个文本文件，这样就可以保证数据不会在服务器关机时丢失。Page Counter 组件使用一个内部 Central Management 对象来记录应用程序中每一页被打开的次数。

PageCounter 对象有 3 个方法：Hits、PageHit 和 Reset。其中，Hits 方法显示指定的网页被打开的次数，PageHit 方法使当前页面的访问次数增加一次，Reset 方法将指定的网页的访问次数重置为 0。

【练习 6-6】使用 PageCounter 对象跟踪访问者的个数并发送特定的消息给第 1000 个

访问者，效果如图 6-15 所示。

(1) 创建一个名为 Counter.asp 的文件，其代码如下所示：

```
<%
Set MyPageCounter = Server.CreateObject("MSWC.PageCounter")
HitMe = MyPageCounter.Hits
If HitMe = 1000 Then
%>
  祝贺您，您是第一千个访问者！<BR>
<% Else %>
  欢迎您，你是第#<%= HitMe %>个访问者 <BR>
<% End If %>
<%MyPageCounter.PageHit%>
```

(2) 将 Counter.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Counter.asp 并按 Enter 键，网页效果将如图 6-16 所示。

(4) 当 Counter.asp 页面被第 1000 个访问者打开时，将显示如图 6-15 所示的页面。



图 6-15　第 1000 个访问者打开的页面

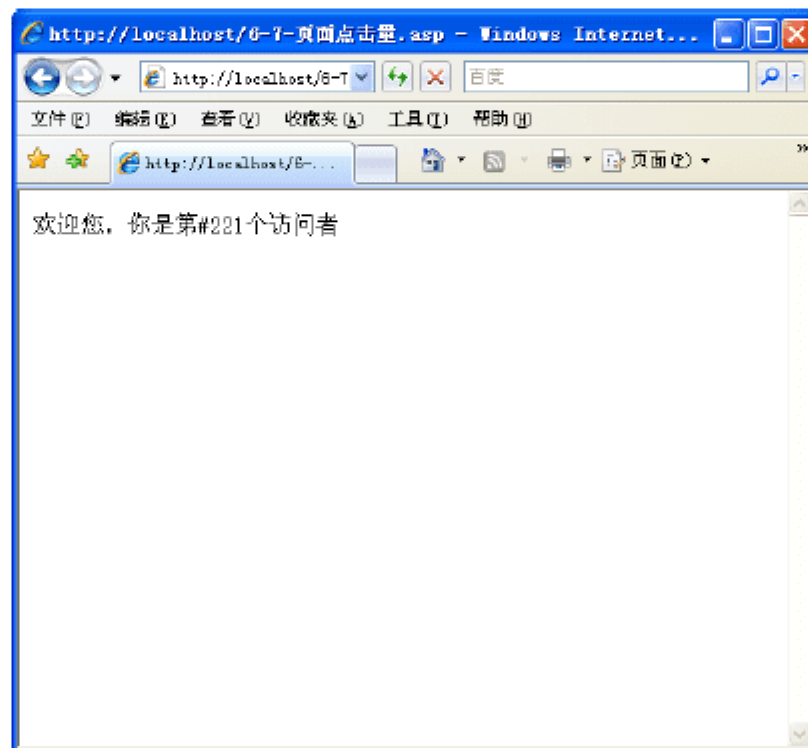


图 6-16　Counter.asp 的效果

6.8　使用 Permission Checker 组件

Permission Checker 组件创建一个 PermissionChecker 对象，该对象使用 IIS 提供的密码验证协议来确认用户是否有权限读取某个文件。PermissionChecker 对象只有一个 HasAccess 方法，该方法决定用户是否有权访问某个指定的文件。如下例所示：

```
<% Set pmck = Server.CreateObject("MSWC.PermissionChecker") %>
Physical Path Access = <%= pmck.HasAccess("c:\pages\abc\default.htm") %>
'该路径是一个网页、文件或资源的物理或虚拟路径
Virtual Path   Access = <%= pmck.HasAccess("/abc/default.htm") %>
```

注意：

可以使用 PermissionChecker 对象来定制网页，这些网页中仅包含当前用户允许访问的资源的链接。这项技术可用来隐藏限制访问的网页和资源，可以使被限制的访问者不知道这些资源的存在。

IIS 支持匿名、基本和集成 Windows 3 类密码验证方法。当启用匿名验证时，用户不需要输入用户名或密码就可以访问站点的公共区域。所有用户初始时都在 IIS 的匿名用户账号下登录，因为匿名用户都享有相同的账号，所以当允许匿名访问时，Permissions Checker 组件将无法确定单个用户的身份。

基本验证提示用户输入用户名和密码，但密码用明文发送，优点在于它是 HTTP 规范的一部分并被大多数浏览器支持，缺点是浏览器使用基本验证是以未加密的形式传输密码的，这样通过监视网络通信，其他人就可以非常容易地用某些通用工具来截取和破解密码。

集成 Windows 验证是一种安全的验证，因为用户名和密码不通过网络发送，用户浏览器通过与服务器进行密码交换来证明其知晓密码。

使特定的 ASP 网页拒绝匿名访问有两种方法：一种是在 Windows 资源管理器中，右击相应的 ASP 文件，从弹出的快捷菜单中选择“属性”命令，再在“安全”选项卡中设置该文件允许哪些用户访问；另一种是在 ASP 脚本中检查环境变量 LOGON_USER 是否为空，如果为空，表示当前用户是匿名用户。

6.9　使用 MyInfo 组件

MyInfo 组件可创建一个 MyInfo 对象，该对象跟踪 Web 站点所有者的个人信息，如经营 Web 站点的组织名称、地址及 Web 站点的设置，Web 站点是否有 guest book。

每个站点只能创建一个 MyInfo 对象，只需创建一个简单的 MyInfo 组件的实例就可以使 Web 网站上的所有页面均能获得其信息。通常是在 Global.asa 文件中用以下命令来创建 MyInfo 对象：

```
<OBJECT RUNAT=Server SCOPE=Session ID=MyInfo PROGID="MSWC.MyInfo">
</OBJECT>
```

创建 MyInfo 对象后，它的属性值保存在文本文件 myinfo.xml 中。myInfo.xml 默认的保存位置是 C:\WINDOWS\system32\inetsrv。

MyInfo 组件默认没有属性和方法，用户可根据需要自行添加。如下例：

```
<%
objMyInfo.MyManager = "zhangshihua"
objMyInfo.MyPhoneExtension = "03766123456"
%>
```

属性添加完成后，可用别的页面对其进行调用和取值。这种不需要 Session 对象在两个页面请求之间存储值的方法，为将来可能改变的数值提供了一个存储区域。采用这些值的页面将会在下次运行时自动地采集这些变化的值，从而避免必须编辑大量的其他页面。如下例：

```
<%
strManagerName = objMyInfo.MyManager
strPhoneExtension = objMyInfo.MyPhoneExtension
%>
```

6.10　使用 Tools 组件

Tools 组件创建一个 Tools 对象，该对象提供的实用程序可使用户很容易地将高级功能加入到 Web 页中。Tools 组件提供了生成随机数，检查文件是否存在或处理 HTML 表格的方法，如表 6-5 所示。

表 6-5　Tools 对象的方法

方　法	说　明
FileExists	检查文件是否存在
Owner	检查当前用户是否是站点所有者
PluginExists	检查服务器插件是否存在(只适用于 Macintosh 计算机)
ProcessForm	处理 HTML 表单
Random	生成一个随机整数

Random 方法产生一个位于－32 768~32 767 的随机整数，为了获得一个指定范围的整数，可以使用脚本语言中的 ABS 函数并对下一个最大的整数取模。

【练习 6-7】使用 Tools 组件生成随机整数，程序运行效果如图 6-17 所示。

(1) 创建一个名为 Tools.asp 的文件，其代码如下所示(如图 6-18 所示):

```
<HTML>
<H2>用 Tools 组件生成随机整数</H2><br>
<Hr>
<%
Set Tools=Server.CreateObject("MSWC.Tools")%>
<%
RandInt=Tools.Random
Response.Write "下面是一个随机生成的整数: "&RandInt&"<Br>"
RandInt=Abs(Tools.Random )
Response.Write "下面是一个随机生成的正整数: "&RandInt&"<Br>"
```

```
RandInt=Abs(Tools.Random )Mod 200
Response.write "下面是一个随机生成的在 0~200 之间的正整数: " &RandInt&"<Br>"
RandInt=(Abs(Tools.Random)Mod 51)+50
Response.write "下面是一个随机生成的在 50~100 之间的正整数: " &RandInt&"<Br>"
%>
<Hr>
-----------------<A Href="tool.asp">刷新</A>-----------------
</HTML>
```

(2) 将 Tools.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Tools.asp 并按 Enter 键，网页效果将如 6-17 所示。



图 6-17　Tools.asp 的效果



图 6-18　Tools.asp 的代码

6.11　使用 IIS Log 组件

IIS Log 组件创建了一个 IISLog 对象，该对象使应用程序能从 IIS 日志文件中抽取特定类型的信息。当用户是服务器上的系统管理员和操作员时，IIS Log 组件特别有用。可以使用下面的脚本来创建 IISLog 对象：

```
Set objLog=Server.CreateObject(MSWC.IISLog)
```

IISLog 对象可提供的方法如表 6-6 所示。

表 6-6　IISLog 对象的方法

方　　法	说　　明
AtEndOfLog	确定所有记录是否已从日志文件中读出
CloseLogFiles	关闭所有打开的日志文件
OpenLogFile	为读或写打开一个日志文件

(续表)

方　法	说　明
ReadFilter	从日志文件中读取指定日期和时间范围的记录
ReadLogRecord	从当前日志文件中读取下一个可用的 Log 记录
WriteLogRecord	写一个 Log 记录到当前日志文件中

6.12　使用 Dictionary 对象

Dictionary 对象是保存键和项目对的字典对象，字典中的项目(可以是任何形式的数据)被保存在数组中，每项都与唯一的键相关联。键值用于检索单个项目，通常是整数或字符串，但不能为数组。

Dictionary 对象包含的属性和方法如表 6-7 所示。

表 6-7　Dictionary 对象的属性和方法

属性或方法	说　明
ArrayMode 属性	数组访问模式，允许以下标方式访问，默认为 True
Count 属性	只读，返回一个对象中的项目数
Item 属性	在一个 Dictionary 对象中设置或返回所指定 key 的 Item
Items 属性	以数组方式返回对象中的全部项目
Key 属性	只读，根据下标取得关键字的键值
Keys 属性	以数组方式返回对象中的全部键值
Add 方法	向对象中添加一个关键字项目对
Exists 方法	如果对象中存在所指定的关键字则返回 True，否则返回 False
Join 方法	返回一个字符串，此字符串由包含在对象中的所有数据键和项目对连接创建
Load 方法	从指定的数据源装载对象内容
Remove 方法	从一个 Dictionary 对象中删除一个关键字项目对
RemoveAll 方法	从一个对象中删除所有的项目
Save 方法	存储对象中的内容到指定的目标
Sort 方法	排序对象中的数据，可以指定升序和降序
Split 方法	分析给定的字符串，将分解出的子字符串填充到对象中

以数字作为下标来查询 Dictionary 对象时，数字表示按照插入顺序的下标，顺序从 0 开始。如果其中某个数据键被删除，则其后插入的数据键的下标依次前移。Dictionary 对象的键值如果为字符串时，将忽略字符串的大小写。

Dictionary 对象的默认属性为 Item，所以可以直接对 Dictionary 对象进行默认操作。例如 dict("c")与 dict.Item("c")的作用是完全等同的。如下例将插入一些关键字项目对，然后修

改项目 c 为第二个插入的项目。

```
Set dict = CreateObject("NetBox.Dictionary")
dict.Add "a", "Athens"
dict.Add "b", "Belgrade"
dict.Add "c", "Cairo"
dict("c") = dict(1)
```

【练习 6-8】用列表方式显示出字典中的所有项目的值，程序运行效果如图 6-19 所示。

(1) 创建一个名为 Dictionary.asp 的文件，其代码如下所示：

```
<H3>添加字典条目</H3>
<Form Action="<%=Request.ServerVariables("Script_Name")%>">
<Input Type="Text" Name="Key">
<Input Type="Text" Name="value">
<Input Type="submit" Value="添加条目">
</Form>
<Hr>
<%
Dim strToDict
strToDict=Trim(Request.Querystring("key"))
strValue=Trim(Request.Querystring("value"))
If Len(strToDict)>0 Then
'检查字典是否存在
If IsEmpty(Session("MyDict")) Then
Set Session ("MyDict")=Server.CreateObject("Scripting.Dictionary")
End If
Dim myLocalDict
Set myLocalDict=Session("MyDict")
myLocalDict.Add strToDict,strValue      '添加字典条目
End If
%>
<H3>列表字典的内容</H3>
<%
If IsEmpty(Session("MyDict")) Then
Response.Write "字典是空的！ "
Else
Dim aKeys,aValues,i
Set myLocalDict=Session("MyDict")
aKeys=myLocalDict.Keys
aValues=myLocalDict.Items
For i=0 To myLocalDict.Count-1
Response.Write aKeys(i)&"="&aValues(i)&"<Br>"
Next
```

```
End If
%>
```

(2) 将 Dictionary.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Dictionary.asp 并按 Enter 键，网页效果将如 6-19 所示。

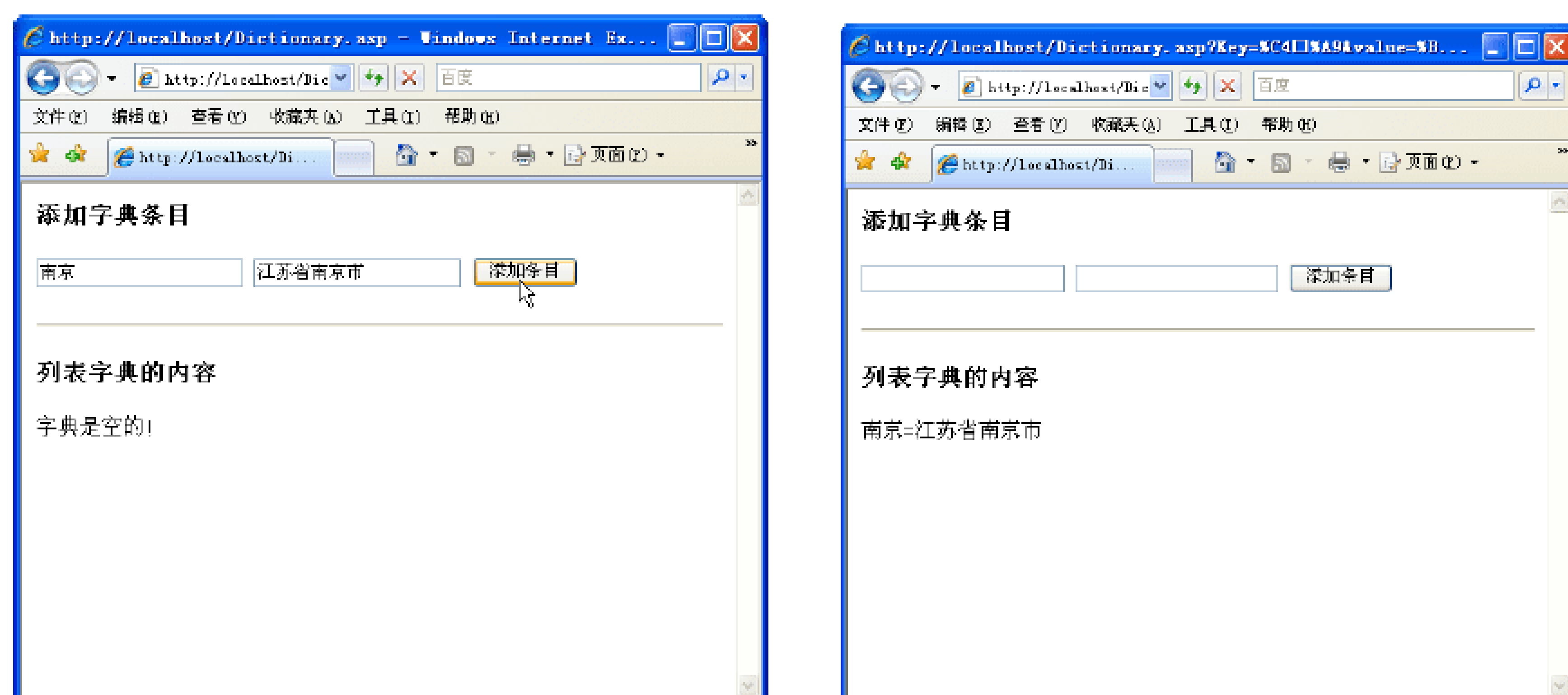


图 6-19　网页效果

6.13　习　　题

6.13.1　填空题

(1) 使用__________组件可快速在网站上建立一个广告系统，它允许在每次访问 ASP 页面时显示新的广告。

(2) 广告条放置到网站后，用户对广告条进行单击操作后，ASP 就会打开__________文件。

(3) ASP 服务器的__________组件通过读取计划文件来完成的网页内容的显示，通常是自动轮换显示一些 HTML 内容。

(4) 在 Content Linker 组件的列表文件中，第一列的 URL 和第二列的链接描述必须用__________键来隔开。

(5) ASP 服务器的__________组件用于检测客户端浏览器的能力。__________组件用来测试访问者对某文件或某页的访问权。

(6) Dictionary 对象的默认属性为__________，所以可以直接对 Dictionary 对象进行默认操作。

6.13.2　选择题

(1) 要使用＿＿＿＿＿＿组件，必须首先创建一个列表文件。

A. Content Linker　　B. Page Counter

C. ScriptTimeout　　D. 以上全是

(2) Page Counter 组件用于创建＿＿＿＿＿＿对象，该对象用来记录和显示 Web 页被打开的次数。

A. Content Linker　　B. Page Counter

C. Page Counter　　D. 以上全错

6.13.3　问答题

(1) ASP 有哪几种调用组件的方法？

(2) 简述使用 Ad Rotator 组件需要哪几个文件，它们都起什么作用？

(3) 试简要说明 Browr Capabilities 组件的工作原理。

6.13.4　操作题

(1) 参照本章练习的操作，使用 Content Linking 组件调用列表文件，实现页面的列表跳转，程序运行效果如图 6-20 所示。

(2) 参照本章练习的操作，创建一个可对字典进行项目查询的页面，程序运行效果如图 6-21 所示。

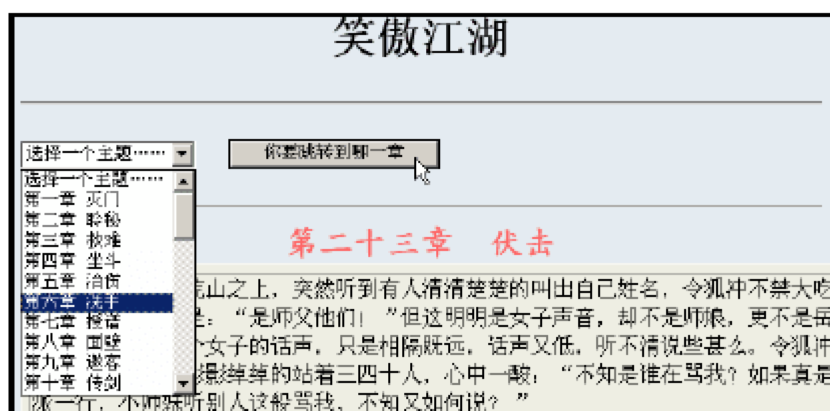


图 6-20　网页效果 1

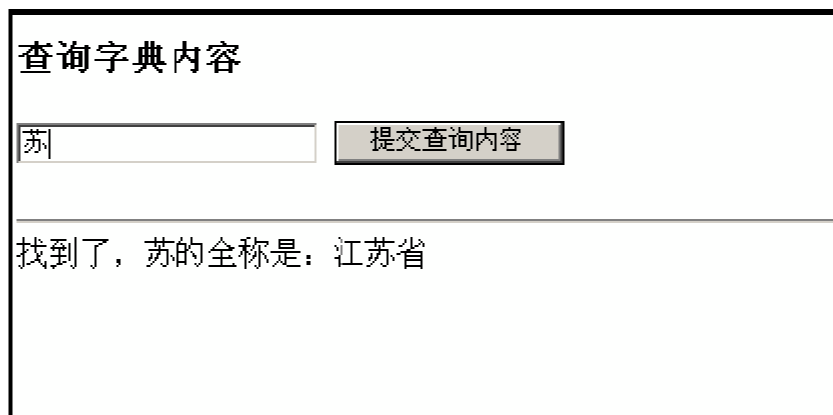


图 6-21　网页效果 2

第7章　Application和Session对象

Application 和 Session 对象是 ASP 重要的内建对象，其中 Application 对象可以在所有用户之间共享信息，并在服务器运行期间持久地保存数据。Session 对象更接近于普通应用程序中的全局变量，全局变量在程序执行的过程中始终有效，其他用户同时启动该程序的另一个副本，该程序的两个实例使用各自的全局变量，在两个进程之间不能互相访问。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 了解 Application 对象的属性
- 掌握 Application 对象的方法
- 掌握 Application 对象的事件
- 掌握 Session 对象的方法
- 掌握 Session 对象的事件

7.1　Application 对 象

Application 对象就是在硬盘上的一组主页以及 ASP 文件。当 ASP 加入了一个 Application 对象，那么它就拥有了作为单独主页所无法拥有的属性。

7.1.1　Application 对象简介

Application 对象是应用程序级的对象，可以产生一个全部 Web 应用程序都可以存取的变量，所有的客户都可以访问这个变量。

应用程序是驻留在 Web 站点的特定目录中的一组文件。每个 Web 站点上可以有多个应用程序，还可以根据某个任务为一些 ASP 文件创建一个应用程序。例如，创建一个应用程序为全部客户服务后，再创建一个新的应用程序为网络管理员服务。

应用程序的运行实例用 Application 对象表示，其生存期从请求该应用程序的第一个页面开始(不是从服务器启动开始)，直到 Web 站点关闭时结束。由于存储在 Application 对象中的数据可以被应用程序的所有用户共享，因而 Application 对象特别适合在应用程序的不同用户之间传递信息。

Application 对象本身提供了一些方法与集合，用以处理 ASP 应用程序的各种状态与特性，如表 7-1 所示。

表 7-1　Application 对象成员

对 象 成 员		说　　明
集合	Contents	存储 Application 对象变量值
	StaticObjects	存储 Application 对象标记<Object>变量
方法	Contents.Remove(valName)	移走 Contents 集合中的特定元素
	Contents.RemoveAll	移走 Contents 集合中的所有元素
	Lock()	锁定 Application 对象存取
	Unlock()	释放被锁定的 Application 对象
事件	OnStart	第一次启动 ASP 应用程序时被触发
	OnEnd	ASP 应用程序结束时被触发

注意：

一个应用程序的根目录由 IIS 的 Internet 信息服务程序来设定，根目录下的每个文件和目录都属于该应用程序。应用程序和应用程序之间是不能互相重叠的，如果其中的某一个子目录也被创建为一个根目录，那它将被认为是一个新的应用程序。

7.1.2　Application 对象的属性

Application 对象没有内置属性，用户可根据需要自行创建。创建一个属性后，在整个应用程序运行期间，此属性的值都可以被所有的用户访问。如下面的代码定义了 Application 对象的属性 Welcome：

```
<% Application("Welcome")="本网站属于非赢利性商业网站" %>
```

每个 Application 变量都是 Contents 集合中的一个成员。创建一个新的 Application 变量，其实就是在 Contents 集合中添加一项新的内容。例如，上面一句代码也可以写为：

```
<% Application.Contents("Welcome")="本网站属于非赢利性商业网站" %>
```

Contents.Remove 方法可以从 Contents 集合中删除一个成员，而 Contents.RemoveAll 方法可以删除 Contents 集合中的所有成员，如下例所示：

```
<% Application.Contents.Remove("Welcome") %>
```

Contents 集合包含了所有的 Application 变量，可使用 For…Each 循环全部列举或显示 Application 变量的值，如下例所示：

```
<%
For Each item in Application.Contents
Response.Write ("<Br>"&item&"="&Application.Contents(item))
Next
%>
```

7.1.3　Application 对象的方法

Application 对象对于网站所有的网页和在线用户来说，是一个公开的对象，因此，任何人在任何时间都有可能对其进行存取。当一个 Application 对象在同一个时间被存取，此时就会发生冲突，因此，对于 Application 对象，适当地限制用户存取非常重要。

Application 对象包含 Lock 和 UnLock 两种方法。Lock 方法用于锁定 Application 对象，保证同一时刻只有一个用户可以操作其中的数据，避免多个用户同时修改同一数据而产生的冲突。UnLock 方法用于解除 Lock 方法对数据的锁定，以便其他用户能访问和修改 Application 对象的属性。Lock 和 UnLock 这两种方法总是成对出现的，这样可以确保 Application 对象中数据对所有用户的完整性和一致性。

【练习 7-1】制作一个网站计数器，程序运行效果如图 7-1 所示。

(1) 创建一个名为 Application.asp 的文件，其代码如下所示(如图 7-2 所示):

```
<%
Application.Lock                    '锁定 Application 对象，保证只有当前用户对计数器进行操作
Application("user_num")= Application("user_num") + 1     '计数器值加 1
Application.UnLock                  '解除锁定，其他用户可对计数器进行操作
Response.write("此页面已经被访问过" &Application("user_num")&"次")     '输出结果
%>
```

(2) 将 Application.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器，然后在地址栏中输入 http://localhost/Application.asp 并按 Enter 键，运行效果如图 7-1 所示。

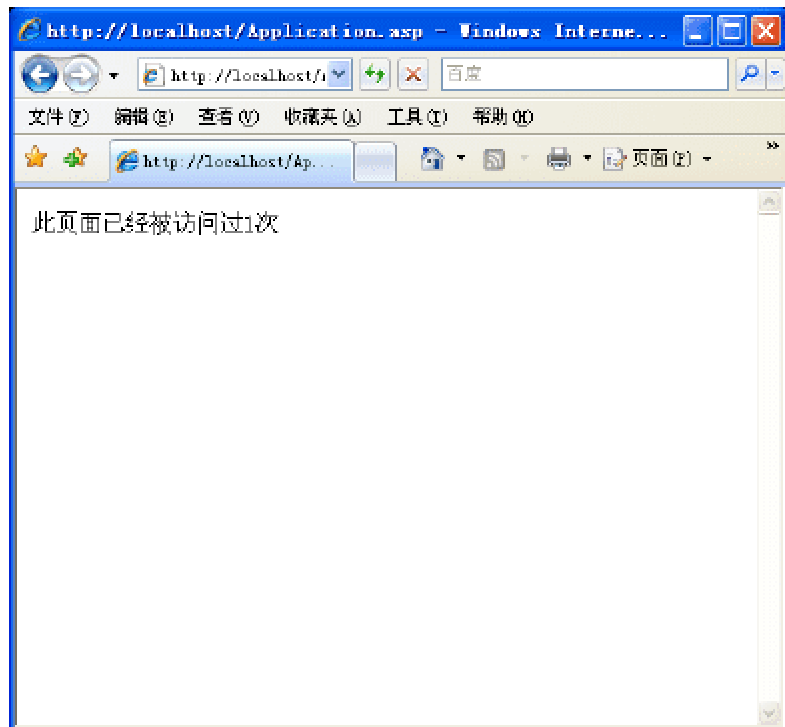


图 7-1　Application.asp 的效果

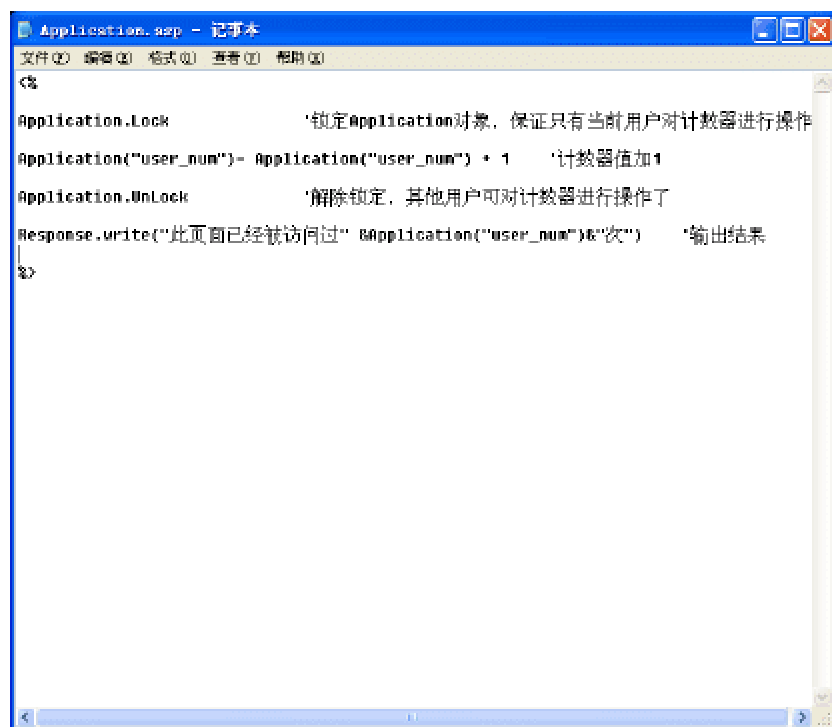


图 7-2　Application.asp 的代码

7.1.4　Application 对象的事件

Application 对象包含 Application_OnStart 和 Application_OnEnd 两个事件。当网站的第一个用户通过浏览器打开网页时，Application 对象就会被自动创建，与此同时，Application

对象所定义的 OnStart 事件便会被触发，对于 ASP 网页的程序设计师而言，会将焦点放在所触发的 OnStart 事件，整个网站一开始所要执行的工作，初始化操作的相关程序代码，都可以在这个事件中作处理。

同样，ASP 应用程序网站被关闭时，系统检测到最后一个用户离线的同时，Application 对象就会被结束。所有 ASP 应用程序执行期间，网页在其中所存储的数据均会被清除。此时，OnEnd 事件即会被触发，处理 ASP 网站结束时所需的程序代码可以放在这个事件中作处理。

处理 OnStart 和 OnEnd 事件的相关程序不同于一般的方法，必须在一个名为 Global.asa 的特殊文件中作处理。Application 对象创建与消失时，都会在该文件内进行(关于 Global.asa 文件的详细说明请参见第 5.4 节中的相关内容)。

7.1.5　利用 Application 对象保存数据

在 Application 对象中，保存的内容除了可以是简单数据类型的变量和普通对象外，还可以是一个保存有多个值的数组。例如，在创建的过程中要求定义一个普通的数组并对其赋值，然后将数组整体定义为一个 Application 对象，代码如下所示：

```
<%
dim Array()
Array=Application("array")
for i = 0 to ubound(array)
  Response.write Array(i)
next i
%>
```

数组在 Application 中只能作为一个对象保存，用户只能对一个数组整体进行存取操作，而不能直接改变数组中某个元素的值。对 Application 对象中数组值的修改也要通过普通的数组来进行。

对一个 Application 对象中的数组值进行修改，其代码如下所示：

```
<%
dim Array()                    '定义一个临时数组
Array=Application("array")     '把含有数组值的 Application 对象赋给该临时数组
Array(0)="第一个元素的值"      '根据需要修改这个数组中元素的值
Array(1)="第二个元素的值"
Application.lock
Application("array")=Array     '最后把数组赋给 Application 对象
Application.unlock
%>
```

7.2　Session 对 象

使用 Session 对象，可以存储特定用户会话所需的信息。当用户在应用程序的页面之间跳转时，存储在 Session 对象中的数据始终存在，不会清除。

7.2.1　Session 对象简介

Web 上用在浏览器和服务器之间传送请求和响应的 HTTP 协议是无状态协议，Web 服务器将每个页面请求都当作独立的请求，服务器不保留以前请求的任何信息。

ASP 的 Session 对象弥补了 HTTP 无法记忆先前请求的缺陷。Session 对象可用来标识每次访问的用户并收集信息，用户在应用程序的页面之间跳转时，该 Session 信息仍然存在并保持不变。在用户与网站服务器保持联系期间，应用程序可调用这些存储的信息来跟踪用户的喜好或选择。

Session 对象同样提供了多种成员，其中包含属性、方法和事件，这些成员在应用程序中的使用也相当广泛，如表 7-2 所示。

表 7-2　Session 对象成员

对 象 成 员		说　明
集合	Contents	存储 Session 对象变量值
	StaticObjects	存储 Session 对象<Object>变量
方法	Contents.Remove(valName)	移走 Session 集合中的特定元素
	Contents.RemoveAll	移走 Session 集合中的所有元素
	Abandon()	结束当前的 Session，为用户创建一个新的 Session
事件	onStart	一个新的用户联机时被触发
	onEnd	一个用户结束联机时被触发
属性	CodePage	设定网页所使用的字符编号
	LCID	存取网页设定的区域识别
	SessionID	代表一个特定用户的唯一 Session 识别 ID
	TimeOut	设定 Session 对象的存活时间

Session 与 Application 对象的应用范围不同，但观念、相关方法以及事件的处理机制均相同。Session 对象用于记载单个客户的信息，Web 服务器为每个访问者建立一个单独的 Session，例如 Session 对象可记载该客户的用户名称及个人爱好等。

Application 对象可以记载所有的客户信息，例如 Application 对象应用于聊天室，大家的发言都存放到 Application 对象中，彼此可以看到所有的发言内容。不同的客户必须访问

属于自己的 Session 对象，但可以访问公共的 Application 对象。

7.2.2　Session 对象的 SessionID 属性

用户第一次请求应用程序中的 ASP 文件时，ASP 将生成一个 SessionID。SessionID 是通过复杂算法产生的长整型数据，返回用于当前会话的唯一标识符。新会话开始时，它将自动为每一个 Session 分配不同的编号，服务器将 SessionID 作为 Cookies 存储到用户 Web 浏览器中。

Web 浏览器创建 SessionID 的 Cookies 后，用户请求其他 ASP 文件或请求其他应用程序中运行的 ASP 文件，ASP 会一直调用该 Cookie 来跟踪会话。如果用户放弃了会话或在会话超时后继续请求其他 ASP 文件，ASP 仍将调用同一 Cookie 开始新的会话。只有 Web 服务器重新启动时才清除存储在内存中的 SessionID 设置，或用户重新启动浏览器时才可能收到新的 SessionID Cookies。如下例将取得当前用户的 SessionID 值：

```
<%Response.Write ("Hi，您的 SessionID 自动编号是 <b>" & SessionID. SessionID & " </b> ")%>
```

7.2.3　Session 对象的 TimeOut 属性

TimeOut 属性定义了应用程序的 Session 对象的时限。如果用户在 TimeOut 规定的时间内没有请求或刷新应用程序中的任何页，Session 对象就会自动终止。默认情况下，服务器只保留 Session 对象 20 分钟。

对于特定的会话，如果要设置低于默认应用程序的超时间隔，可通过 TimeOut 属性来设置。TimeOut 属性以分钟为单位指定超时间隔，如下例将超时间隔设为 10 分钟：

```
<% Session.TimeOut=10 %>
```

注意：

可以在 IIS 的 Internet 信息服务程序中设置会话超时，该值的设置取决于应用程序的要求及服务器的内存容量。会话超时如果设置过长，可能会导致打开的会话太多，这将增加服务器内存资源的负担。

7.2.4　Session 对象的 Abandon 方法

Abandon 方法是 Session 对象的唯一方法，Abandon 方法可以用来删除用户的 Session 对象并释放其所占用的资源。下面的语句将用于删除 Session 对象：

```
<% Session.Abandon %>
```

如果使用了 Abandon 方法，Session 对象将被重新分配一个新的 SessionID 值。

7.2.5　Session 对象的事件

Session 对象包含 Session_OnStart 和 Session_OnEnd 两个事件。当网站上一个新用户通过浏览器请求一份网页时，这个用户的专属 Session 对象就会被创建。Session 对象所定义的 OnStart 事件同时被触发。

当一个用户离线或是停止任何浏览网页操作时，一旦过了 Session 对象的存活期限，代表此用户的 Session 对象就会被结束。此时 OnEnd 事件即会被触发，处理用户离线时所需的程序代码可以放在这个事件中作处理。

OnStart 和 OnEnd 事件的相关程序与 Application 对象一样，都是在 Global.asa 文件中作处理的。

7.2.6　利用 Session 对象记录用户登录信息

下面以一个简单的实例，介绍利用 Session 对象记录变量内容的方法。

【练习 7-2】利用 Session 对象记录变量内容，实现 ASP 网页记录用户登录信息的效果。

(1) 创建一个名为 Application3.asp 的文件，其代码如下所示(如图 7-3 所示):

```
<%
data=Session("data")
Response.Write"进入网页时,data="&data&"<BR>"
data=data+1
Response.Write"网页结束时,data="&data&"<BR>"
Session("data")=data
%>
```

(2) 将 Application3.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Application3.asp 并按 Enter 键，效果如图 7-4 所示。

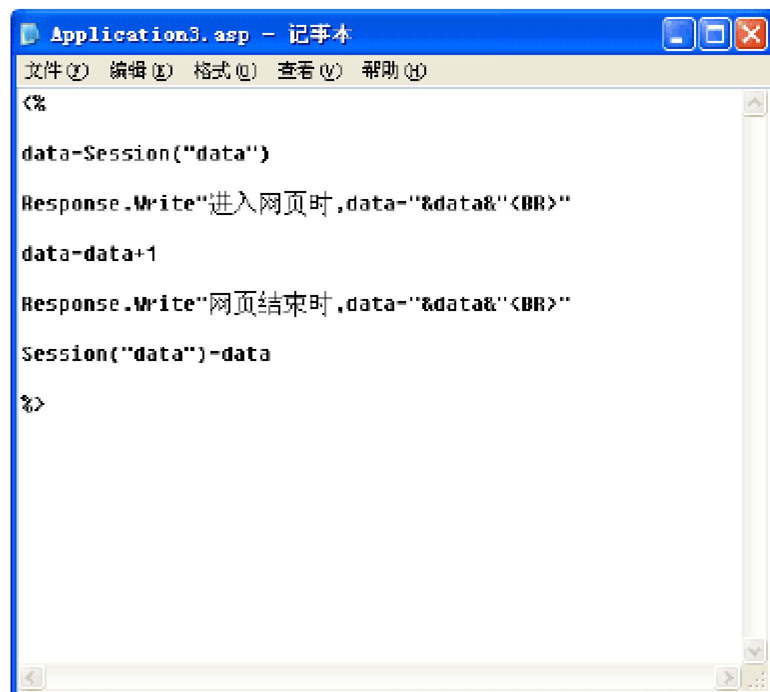


图 7-3　Application3.asp 的代码

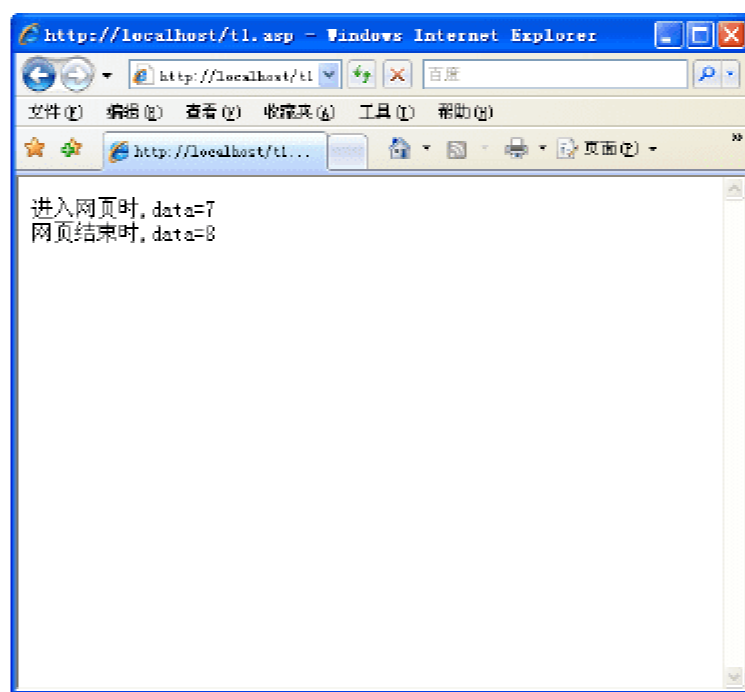


图 7-4　Application3.asp 的效果

7.3　Global.asa 文件

Global.asa 文件是用来存放 Application 对象和 Session 对象事件的程序，当 Application 对象和 Session 对象第一次被调用或结束时，服务器就读取该文件并进行相应的处理。

Global.asa 文件是一个文本文件，可使用任何文本编辑器进行编辑。下面就是一个标准的空白 Global.asa 文件结构：

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
Sub Application_OnStart
'当第一个用户运行 ASP 应用程序中的任何一个页面时执行
End Sub
Sub Application_OnEnd
'当 Web 服务器关闭时执行
End Sub
Sub Session_OnStart
'用户第一次运行 ASP 应用程序中的任何一个页面时执行
End Sub
Sub Session_OnEnd
'当一个用户的会话超时或退出应用程序时执行
End Sub
</SCRIPT>
```

ASP 对使用 Global.asa 文件有以下几条要求：

- 每一个应用程序可能由很多文件或文件夹组成，但只能有一个 Global.asa 文件，而且文件名必须为 Global.asa。
- 必须存放在应用程序的根目录中。
- Global.asa 文件不能写成<%…%>的形式，如果包含的脚本没有用<SCRIPT>标记封装，或定义的对象没有会话或应用程序作用域，则服务器将返回错误。服务器会忽略已标记的但未被应用程序或会话事件使用的脚本和文件的 HTML 语句。
- 在 Global.asa 文件中不能包含任何输出语句，如 Response.Write，因为 Global.asa 文件只是被调用，而不会显示在页面上。

【练习 7-3】通过 Global.asa 文件，使用 Application 对象和 Session 对象显示网站的在线人数和访问人数，程序运行效果如图 7-5 所示。

(1) 创建一个名为 Global.asa 的文件，其代码如下所示：

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
Sub Application_OnStart
Session.TimeOut=3        '将会话超时设为 3 分钟
Application.Lock        '锁住 Application
```

```
Application("intuseronline")=0    '初始化在线人数为 0
Application("intuserall")=0      '初始化访问人数为 0
Application.UnLock         '解开 Application
End Sub
Sub Session_OnStart
Application.Lock
'当一次新用户会话开始时，使在线人数和访问人数都加 1
Application("intuseronline")=Application("intuseronline")+1
Application("intuserall")=Application("intuserall")+1
Application.UnLock
End Sub
Sub Session_OnEnd
Application.Lock
'会话超时或用户退出时，使在线人数减 1
Application("intuseronline")=Application("intuseronline")-1
Application.UnLock
End Sub
</SCRIPT>
```

(2) 将 Global.asa 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 创建一个显示访问人数的 ASP 页面 Application4.asp，其代码如下所示(如图 7-6 所示):

```
<h2 align="center">网站的统计数据</h2>
<%
Response.Write "<Br>当前网站在线人数是："&Application("intuseronline")
Response.Write "<Br>网站总的访问量是："&Application("intuserall")
%>
```

(4) 将 Application4.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(5) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Application4.asp 并按 Enter 键，效果如图 7-5 所示。



图 7-5　Global.asa 的效果

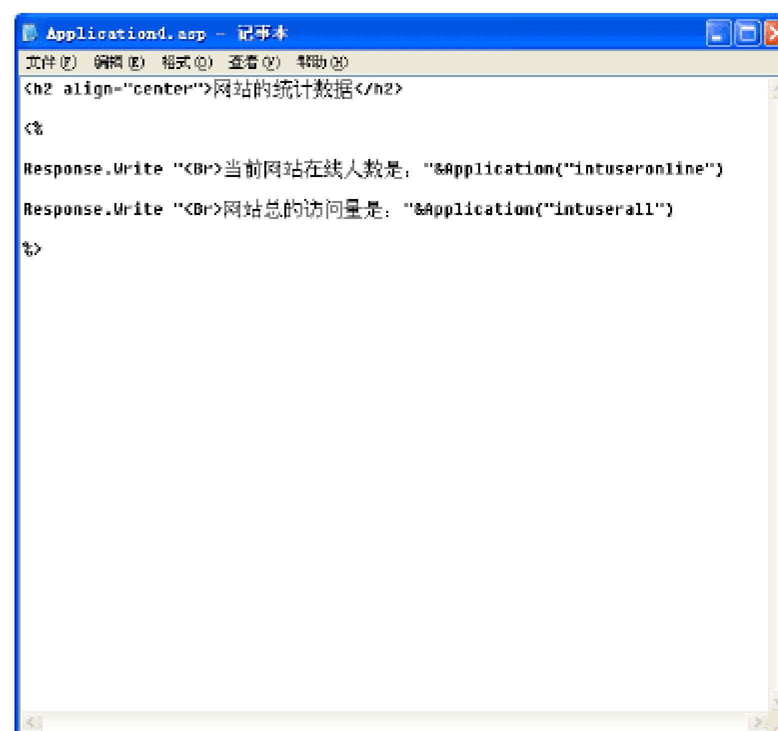


图 7-6　Global.asa 的代码

7.4　习　　题

7.4.1　填空题

(1) ASP 提供的 6 个内置对象分别是___________、___________、___________、___________、___________和 ScriptError 对象。

(2) 每个 Application 变量都是___________集合中的一个成员。

(3) ___________方法用于锁定 Application 对象，___________方法可以解除对 Application 对象的锁定。

(4) 默认情况下，服务器只保留 Session 对象___________分钟。

(5) 如果使用了___________方法，Session 对象将被重新分配一个新的 SessionID 值。

(6) ___________文件是用来存放 Application 对象和 Session 对象事件的程序。

7.4.2　选择题

(1) 下面(　　)不是 Global.asa 文件可用来存放的事件。

A. Application_OnEnd　　B. Application_OnStart

C. Sever_OnStart　　D. Session_OnEnd

(2) 下面关于脚本延时的时间期限，说法不正确的是(　　)。

A. 脚本运行超过 ScriptTimeout 属性设置时间将作超时处理

B. 超时处理时会中止没有完毕的响应并提示超时错误信息

C. 如果设置的 ScriptTimeout 属性值小于默认值，则以设置值为准

D. 脚本可以设置为永不过期

7.4.3　问答题

(1) 简述 Session 对象与 Application 对象的异同。

(2) ASP 对使用 Global.asa 文件有哪些要求？

7.4.4　操作题

(1) 编辑一段查询当前 SessionID 属性值的 ASP 程序，以显示如图 7-7 所示的网页效果。

(2) 参照本章所介绍的内容，编写一个 Global.asa 文件，以显示如图 7-8 所示的网页效果。



图 7-7　网页效果 1

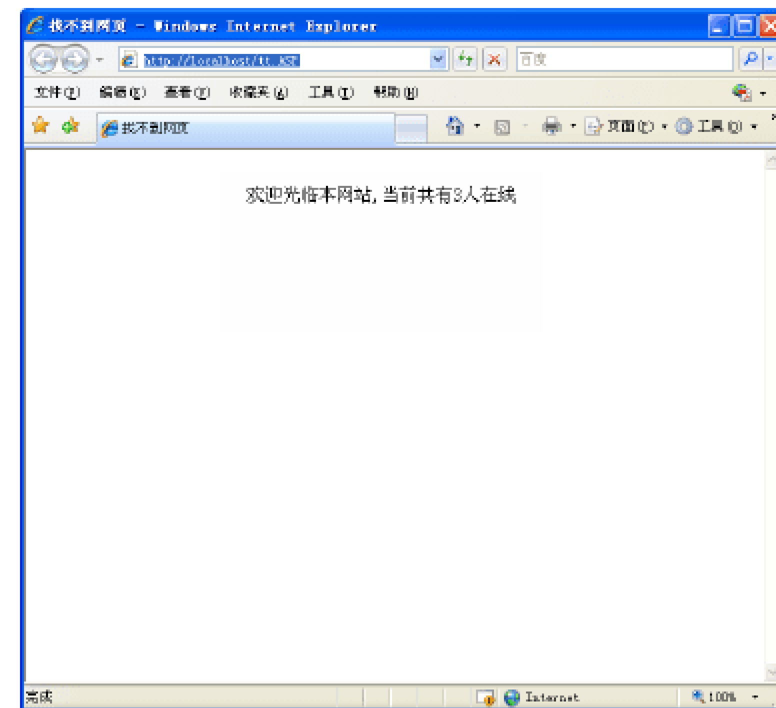


图 7-8　网页效果 2

第8章　ASP的文件处理

ASP 的 File Access 组件提供了可用来访问计算机文件系统的方法和属性。File Access 组件使用 FSO(File System Object)对象模型来处理服务器端的文件、文件夹和驱动器。使用 FSO 可以建立、删除、移动文件和文件夹，检查文件和文件夹是否存在，以及更改文件和文件夹的名字。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 了解 FileSystemobject 对象
- 掌握在文件中读写数据的方法
- 掌握文件的各种操作
- 了解获得驱动器信息的方法
- 了解文件夹的操作

8.1　FileSystemObject 对象简介

File Access 组件提供可用于在计算机文件系统中检索和修改文件的对象。用户可以使用 File Access 组件创建 FileSystemObject 对象，简称为 FSO。FSO 提供用于访问文件系统的方法、属性和集合。

FSO 可以对驱动器、文件夹以及文件等对象进行相关的处理操作。可以检测并显示出系统驱动器的信息分配情况；可以创建、改变、移动和删除文件夹，并能检测一些给定的文件夹是否存在；可以提取出文件夹的信息，如名称、被创建或最后一次修改的日期等。

注意：

FSO 对象模式使文件处理变得很容易。当只想存取一些更新更小的数据时，可把它们存储为二进制或文本文件中，然后用 FSO 来创建文件，插入和改变数据，以及输出(读取)数据。数据存储在数据库中，如 Access 或 SQL 服务器，会给应用程序增加很大的开销。而使用 FSO 将数据存储在文本文件中可节省系统资源。

FSO 组件中包含多个对象和集合，其名称和相关说明如表 8-1 所示。

表 8-1　FileSystemObject(FSO)包含的对象和集合

对象/集合	描　　述
FileSystemObject	主对象。包含用来创建、删除和获得有关信息，通常用来操作驱动器、文件夹和文件的方法和属性。和该对象相关联的许多方法，与其他 FSO 对象中的方法完全相似，它们是为了方便才被提供的
Drive	对象。包含用来收集信息的方法和属性，这些信息与连接在系统上的驱动器有关，如驱动器的共享名以及有多少可用空间。请注意，Drive 并非必须是硬盘，也可以是 CD-ROM 驱动器，RAM 磁盘等。并非必须把驱动器实物地连接到系统上，也可以通过网络在逻辑上被连接起来
Drives	集合。提供驱动器的列表，这些驱动器可以是物理的或在逻辑上与系统相连接。Drives 集合包括所有驱动器，与类型无关。要可移动的媒体驱动器在该集合中显现，不必把媒体插入到驱动器中
File	对象。包含用来创建、删除或移动文件的方法和属性，也用来向系统询问文件名、路径和多种其他属性
Files	集合。提供包含在文件夹内的所有文件的列表
Folder	对象。包含用来创建、删除或移动文件夹的方法和属性，也用来向系统询问文件夹名、路径和多种其他属性
Folders	集合。提供在 Folder 内的所有文件夹的列表
TextStream	对象。用来读写文本文件

注意：

与 FileSystemObject 主对象相关联的很多方法重复了另外 4 个对象的方法。因此既可以通过 FileSystemObject 主对象来对驱动器、文件夹和文件进行大多数操作，也可以通过对应的驱动器、文件夹或文件对象对这些组件进行操作。FSO 模型通过两种方法实现对同一对象的操作，其操作效果是相同的，提供这种冗余功能的目的是为了实现最大的编程灵活性。

使用 FSO 编程，首先要用 Server.CreatObject 方法创建 FileSystemObject 对象，如下例所示：

```
<%
Dim MyFileObject
Set MyFileObject =Server.Create Object("Scripting.FileSystemObject")
%>
```

创建 FileSystemObject 对象后，就可以创建、打开或读写文件，以及对文件和文件夹进行新建、复制、移动、删除等操作。FileSystemObject 对象的方法及相关说明如表 8-2 所示。

表 8-2　FileSystemObject 对象的方法

方　法	说　明
CreateTextFile	创建一个文本文件
OpenTextFile	打开一个已有的文本文件
GetFile	返回与指定路径中某文件相应的 File 对象
GetExtensionName	获得文件扩展名
CopyFile	复制一个文件
MoveFile	移动文件，将文件从一个路径移到另一路径
DeleteFile	删除文件，其中第一个参数是要删除的文件，第二个参数说明是否强行删除(如果文件是只读或隐含等)
FileExists	如果指定的文件存在，返回 True，否则返回 False
GetFolder	返回与指定的路径中某文件夹相应的 Folder 对象
CreateFolder	创建一个目录
CopyFolder	复制整个目录，会复制这个目录下的所有文件
MoveFolder	移动目录，会移动目录下包含的全部文件
DeleteFolder	删除目录，会删除目录下所有文件，并且删除的文件不会进入回收站
FolderExists	如果指定的文件夹存在，返回 True，否则返回 False

8.2　在文件中读写数据

FSO 功能最强大的部分就是对文件的操作，可以用来记数、内容管理、搜索，还可以用来生成动态 HTML 页面等。

新建文本文件需要使用 FileSystemObject 对象和该对象的 CreateTextFile 方法。TextStream 对象提供对存储在磁盘上文件的访问，能够读出或写入顺序文本文件。

8.2.1　TextStream 对象的方法和属性

TextStream 对象必须通过 FileSystemObject 对象进行实例化，所以可以把 TextStream 对象当作是 FileSystemObject 对象的子对象。TextStream 对象的方法及其相关说明如表 8-3 所示。

表 8-3　TextStream 对象的方法

方　法	说　明
Close	用来关闭一个已打开的数据流文件和其对应的文本文件
Read	从打开的文本文件中读取从光标位置开始的指定数目的字符

(续表)

方　法	说　明
ReadAll	用来读取一个已打开的数据流文件内的所有数据
ReadLine	用来读取一个已打开的数据流文件内的一行数据
Skip	用来跳过已打开的数据流文件内的字符数目
SkipLine	用来跳过已打开的数据流文件内的一整行数据
Write	用来写入数据至一个已打开的数据流文件
WriteLine	用来写入一整行数据至一个已打开的数据流文件
WriteBlankLines	用来指定欲写入的新行数目

TextStream 对象的属性及其相关说明如表 8-4 所示。

表 8-4　TextStream 对象的属性

属　性	说　明
AtEndOfLine	当光标位于当前行的末尾时，其值为 True，否则为 False
AtEndOfStream	当光标位于流的末尾时，其值为 True，否则为 False
Column	计算从行首到当前光标位置的字符数
Line	计算光标所在行在整个文件中的行号

8.2.2　创建文本文件

创建或打开一个文本文件并返回 TextStream 对象可使用 CreateTextFile 方法或 OpenTextFile 方法。

1. CreateTextFile 方法

用 CreateTextFile 方法创建新的文本文件，或覆盖一个已存在的文件返回的 TextStream 对象可用来读写文件，其语法结构如下：

```
Object.CreateTextFile(filename[,overwrite[,unicode]])
```

其中，Object 应为 FileSystemObject 或 Folder 对象的名称。filename 是字符串，用于指明待创建的文件。overwrite 指明是否可覆盖已有的文件，若设为 True，则可以覆盖；若设为 False(默认)，则不能覆盖。unicode 指明是以 unicode 格式还是以 ASCII 格式创建文件，若设为 True，则以 unicode 格式创建；若设为 False(默认)，则以 ASCII 格式创建。

【练习 8-1】使用 CreateTextFile 方法在真实路径指定的位置创建一个空的文本文件。

```
<%
```

```
Dim fso,fil
Set fso=Server.CreateObject("Scripting.FileSystemObject")
Set fil=fso.CreateTextFile("D:\My webs\test1.txt",True)
%>
```

注意：

通过 FSO 打开驱动器、文件夹和文件，只能使用指定的绝对物理路径地址。而文本文件创建的位置常需要使用虚拟路径来指定，因为这样可便于将应用程序转移到不同的计算机上运行，所以对虚拟路径常使用 Server.MapPath()方法将其转化为真实路径。

2. OpenTextFile 方法

用 OpenTextFile 方法可打开一个已有的文本文件，也可创建一个文本文件。它返回一个 TextStream 对象，可用这个对象对文件读或追加数据。

```
Object.OpenTextFile(filename[,iomode[,create[,format]]])
```

其中，Object 应为 FileSystemObject 对象的名称；filename 是字符串，用于指明待创建的文件；iomode 用于指定输入/输出模式，取值为 1(以只读模式打开，不能对文件进行写操作)、2(以只写方式打开，不能对文件进行读操作)或 8(以追加方式打开，可以在文件末尾进行写操作)；create 指出文件不存在时是否创建，默认是 False，若要创建，可设为 True；format 指出以何种格式打开文件，取值是－2(以系统默认格式打开)、－1(以 unicode 格式打开)或 0(以 ASCII 格式打开)。

【练习 8-2】使用 OpenTextFile 方法在虚拟路径指定的位置创建一个空的文本文件。

```
<%
Dim fso,ts
Const ForWriting=2
Set fso=CreateObject("Scripting.FileSystemObject")
Set fs=fso.OpenTextFile(Server.MapPath("test2.txt"),ForWriting,True)
%>
```

8.2.3　添加数据到文本文件

文本文件一经创建，就可以分 3 步向其中加入数据：即打开文件，写入数据，关闭文件。打开文件的方法有两种，第一种是用 File 对象的 OpenAsTextStream 方法，第二种是用 FileSystemObject 对象的 OpenTextFile 方法。

打开文件后，就可以用 TextStream 对象的 Write 或 WriteLine 方法写入数据，两者间的唯一差别是 WriteLine 方法会在字符串的末尾添加换行符。如果想在文本文件中添加一

个空行，可使用 WriteBlankLines 方法。

对文本文件的操作进行完成后，要使用 TextStream 对象的 Close 方法关闭文件。

【练习 8-3】创建一个文本文件，并写入文本和空行，创建后的文本如图 8-1 所示。

```
<%
Dim fso,fil
Set fso=CreateObject("Scripting.FileSystemObject")
Set fil=fso.CreateTextFile(Server.MapPath("test3.txt"),True)
fil.Write("***第一句文本")
fil.WriteLine("***第二句带换行符的文本")
fil.WriteLine("***第三句带换行符的文本")
fil.WriteBlankLines(2)  '写入两个空白行
fil.Write("***第四句文本")
fil.Close  '关闭文件
%>
```

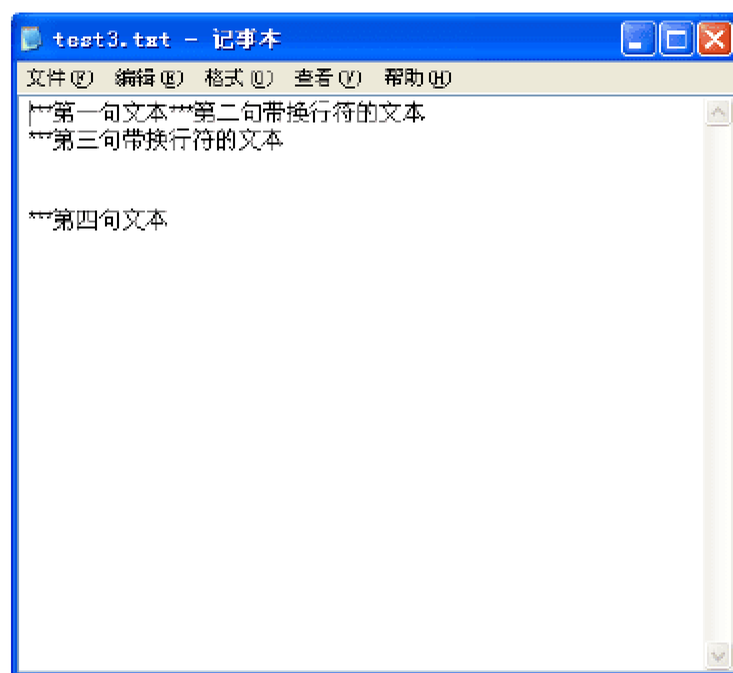


图 8-1　添加数据到文本文件

8.2.4　读取文本文件

要从文本文件中读取数据，可以使用 TextStream 对象的 Read、ReadLine 或 ReadAll 方法。Read 方法从文本文件中读取指定数量的字符；ReadLine 方法读取一整行，但不包括换行符；ReadAll 方法读取一个文本文件的所有内容。

在使用 Read 或 ReadLine 方法时，还可以用 Skip 方法跳过几个字符，或用 SkipLine 方法跳过几行。

【练习 8-4】分别使用 Read、ReadLine 或 ReadAll 方法读取【练习 8-3】所创建的文本文件(test3.txt)的内容，程序运行效果如图 8-2 所示。

(1) 将如图 8-1 所示的 test3.txt 文件复制至 C:\Inetpub\wwwroot 文件夹中。

(2) 创建一个名为 read.asp 的文件，其代码如下所示(如图 8-3 所示):

```
<%
Dim fso,f1,ts,s1,s2
Const ForReading=1
Set fso=CreateObject("Scripting.FileSystemObject")
Set ts=fso.OpenTextFile(Server.MapPath("test3.txt"),ForReading)
s1=ts.Read(3)
Response.Write "文件的前三个字符内容是：'"&s1&"'"
Response.Write "<Br><Br>"
ts.Skip(5)          '从当前位置往后跳过 5 个字符
s2=ts.ReadLine
Response.Write "第一行剩下的内容是：'"&s2&"'"&"<Br><Br>"
Response.Write "文本剩下的内容是：'"&ts.ReadAll&"'"&"<Br>"
ts.Close
%>
```

(3) 将 read.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(4) 启动 IE 浏览器后，在地址栏中输入 http://localhost/read.asp 并按 Enter 键，效果如图 8-2 所示。

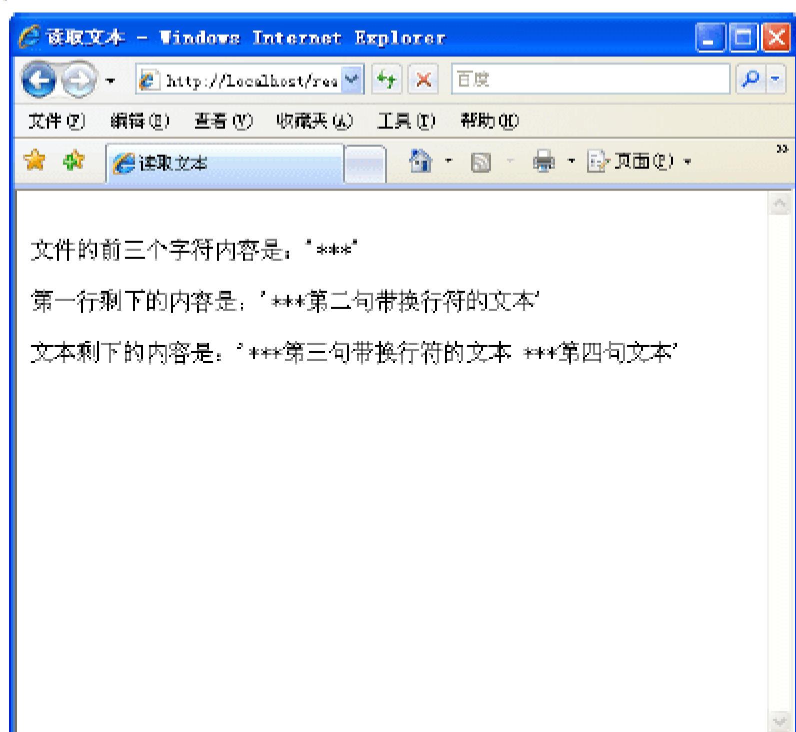


图 8-2　read.asp 的效果



图 8-3　read.asp 的代码

8.3　对文件进行的各种操作

File 对象提供了对文件属性的访问，通过它的方法能够对文件进行操作。File 对象允许复制、删除以及移动文件，还可以检测文件是否存在并查看文件具有什么属性。

8.3.1　File 对象的属性与方法

File 对象的属性及其相关说明如表 8-5 所示。

表 8-5　File 对象的属性及说明

属　　性	说　　明
Attributes	设置或返回文件的系统属性。可以是下列值中的一个或其组合 0(表示普通文件)、1(表示只读文件)、2(表示隐藏文件)、4(表示系统文件)、16(表示文件夹或目录)、32(表示上次备份后已更改的文件)、1024(表示链接或快捷方式)和 2048(表示压缩文件)
DateCreated	返回该文件的创建日期和时间
DateLastAccessed	返回最后一次访问该文件的日期和时间
DateLastModified	返回最后一次修改该文件的日期和时间
Drive	返回该文件所在的驱动器的 Drive 对象
Name	设定或返回文件的名称
ParentFolder	返回该文件的父文件夹的 Folder 对象
Path	返回文件的绝对路径，可使用长文件名
ShortName	返回按照早期 8.3 文件命名约定转换的短文件名
ShortPath	返回按照早期 8.3 文件命名约定转换的短文件绝对路径
Size	返回该文件的大小(字节)
Type	如果可能，返回一个文件类型的说明字符串

每个 Folder 对象提供了一个 Files 集合，包含文件夹中文件对应的 File 对象。还可以直接地从 FileSystemObject 对象中通过使用 GetFile 方法得到一个 File 对象引用。

File 对象的方法允许复制、删除以及移动文件，它有一个使用文本流打开文件的方法。File 对象的方法及说明如表 8-6 所示。

表 8-6　File 对象的方法及说明

方　　法	说　　明
Copy(destination, overwrite)	将这个文件复制到 destination 指定的文件夹。如果 destination 的末尾是路径分隔符(\)，那么认为 destination 是放置复制文件的文件夹。否则认为 destination 是要创建的新文件的路径和名字。如果目标文件夹已经存在且 overwrite 参数设置为 False，将产生错误，默认的 overwrite 参数是 True
Delete(force)	删除这个文件。如果可选择的 force 参数设置为 True，文件即使具有只读属性也会被删除。默认的 force 是 False
Move()	将文件移动到 destination 指定的文件夹
CreateTextFile	用指定的文件名创建一个新的文本文件，并且返回一个相应的 TextStream 对象
OpenAsTextStream (iomode,format)	打开指定文件并且返回一个 TextStream 对象，用于文件的读、写或追加。Iomode 参数指定了要求的访问类型。Format 参数说明了读、写文件的数据格式

给定一个 File 对象后，可以使用 ParentFolder 属性得到包含该文件的 Folder 对象的引

用，用来在文件系统中导航，甚至可以用 Drive 属性获得相应的 Drive 对象的引用，并得到各种 Folder 对象以及所包含的 File 对象。

另外，给定一个 Folder 对象以及对应的 Files 集合后，可以通过遍历该集合检查这一文件夹中的每个文件，还可以使用 File 对象的各种方法以一定方式处理该文件，如复制、移动或删除。

8.3.2　移动、复制和删除文件

移动一个文件可用 File 对象的 Move 方法或 FileSystemObject 对象的 MoveFile 方法，复制一个文件可用 File 对象的 Copy 方法或 FileSystemObject 对象的 CopyFile 方法，删除一个文件可用 File 对象的 Delete 方法或 FileSystemObject 对象的 DeleteFile 方法。

【练习 8-5】创建一个文本文件(如 test4、txt)，并对它进行复制、移动和删除操作。

(1) 创建一个名为 File.asp 的文件，其代码如下所示：

```
<%
Function ShowCode(filename)
    Set fso = Server.CreateObject("Scripting.FileSystemObject")
    Set cnrs = fso.OpenTextFile(filename, 1)
    While Not cnrs.AtEndOfStream
        rsline = cnrs.ReadLine
        rsline = server.HTMLEncode(rsline)
        Response.Write(rsline & "<br>")
    Wend
end Function
%>
<form action="viewcode.asp" method="post">
输入文件名<input type="file" name="filename">
<input type="submit" value="查看源程序">
</form>
<%
file=request.form("filename")
response.write (file & "源程序如下<hr>")
If trim(file)<> "" then
  Call ShowCode(file)
End If
%>
```

(2) 将 File.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/File.asp 并按 Enter 键，效果如图 8-4 所示。

(4) 单击 File.asp 页面中的“浏览”按钮，然后在打开的“选择文件”对话框中选中本章【练习 8-4】创建的 ASP 程序文件 read.asp 后，单击“打开”按钮。

(5) 单击 File.asp 页面中的“查看源程序”按钮后，网页将显示如图 8-5 所示的 ASP 源代码。

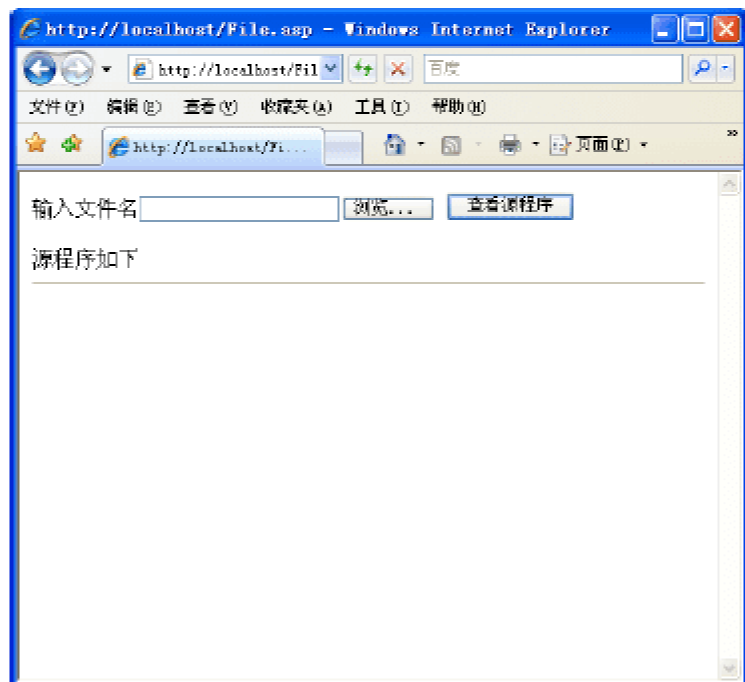


图 8-4　File.asp 的效果

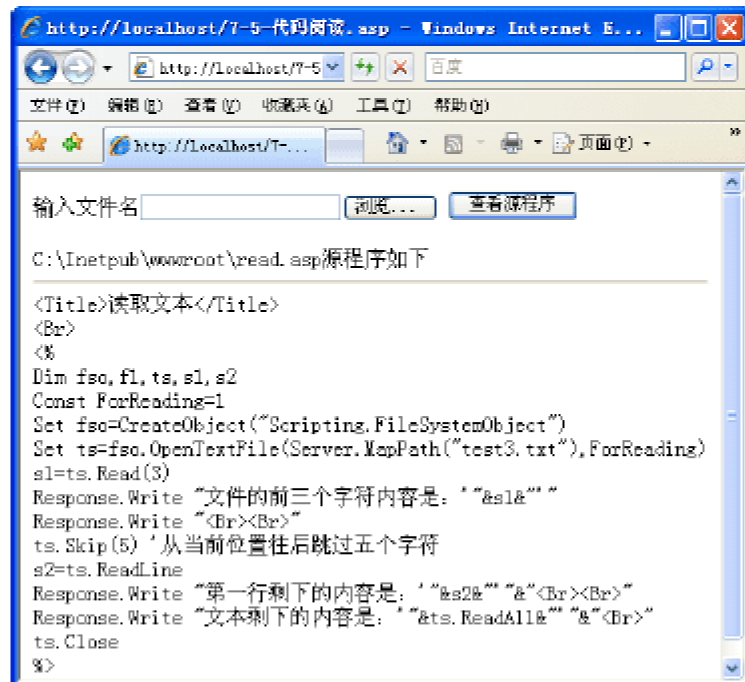


图 8-5　显示程序源代码

8.3.3　检测文件和文件夹是否存在

使用 FileSystemObject 对象的 FolderExists 和 FileExists 方法可以检测文件和文件夹是否存在，若存在，则返回 True，否则返回 False。

【练习 8-6】创建一个 ASP 网页，检测一个指定的文件是否存在，并在网页中返回相应的结果。

(1) 创建一个名为 detection.asp 的文件，其代码如下所示：

```
<%
Dim fso
Set fso=Server.CreateObject("Scripting.FileSystemObject")
If fso.FileExists(Server.MapPath("test1.txt")) Then
Response.Write Server.MapPath("test1.txt")&"存在"
Else
Response.Write Server.MapPath("test1.txt")&"不存在"
End If
%>
```

(2) 将 detection.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/detection.asp 并按 Enter 键，效果如图 8-6 所示。

(4) 创建一个名为 test1.asp 的文本文件后，将其复制至 C:\Inetpub\wwwroot 文件夹中，然后刷新 detection.asp 页面，网页效果如图 8-7 所示。

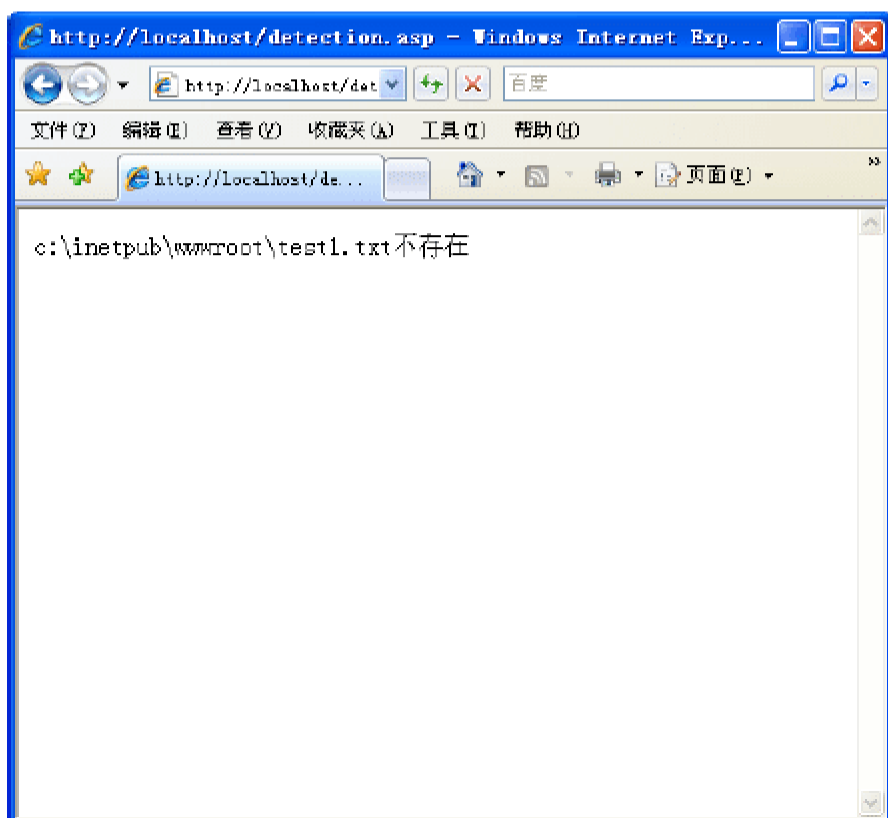


图 8-6　detection.asp 的效果

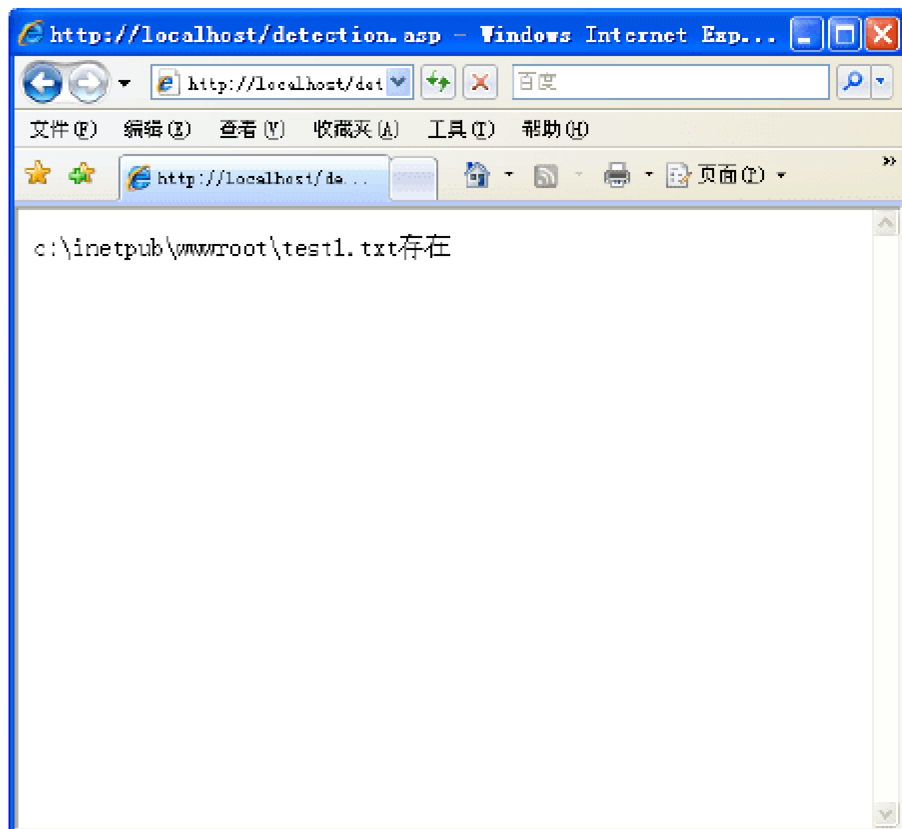


图 8-7　修改后的网页效果

8.3.4　检测文件和文件夹

使用 File 对象和 Folder 对象不仅可以实现文件和文件夹的各种检测操作，而且能得到文件和文件夹的各种属性。

【练习 8-7】创建一个文件，显示计算机 C:根目录下的文件夹，效果如图 8-8 所示。

(1) 创建一个名为 Folder.asp 的文件，其代码如下所示：

```
<%
 dim objfilesys
   dim mydrive
   dim myfolder
   set objfilesys=server.createobject("scripting.filesystemobject")
   set mydrive=objfilesys.getdrive("c")
   set myfolder=mydrive.rootfolder
   set myfolder=myfolder.subfolders
   response.write "c 盘根目录下有"&myfolder.count&"个文件夹<br>"
   for each folder in myfolder
   response.write "子文件夹"&folder.name&"<br>"
   next
   %>
```

(2) 将 Folder.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Folder.asp 并按 Enter 键，效果如图 8-9 所示。

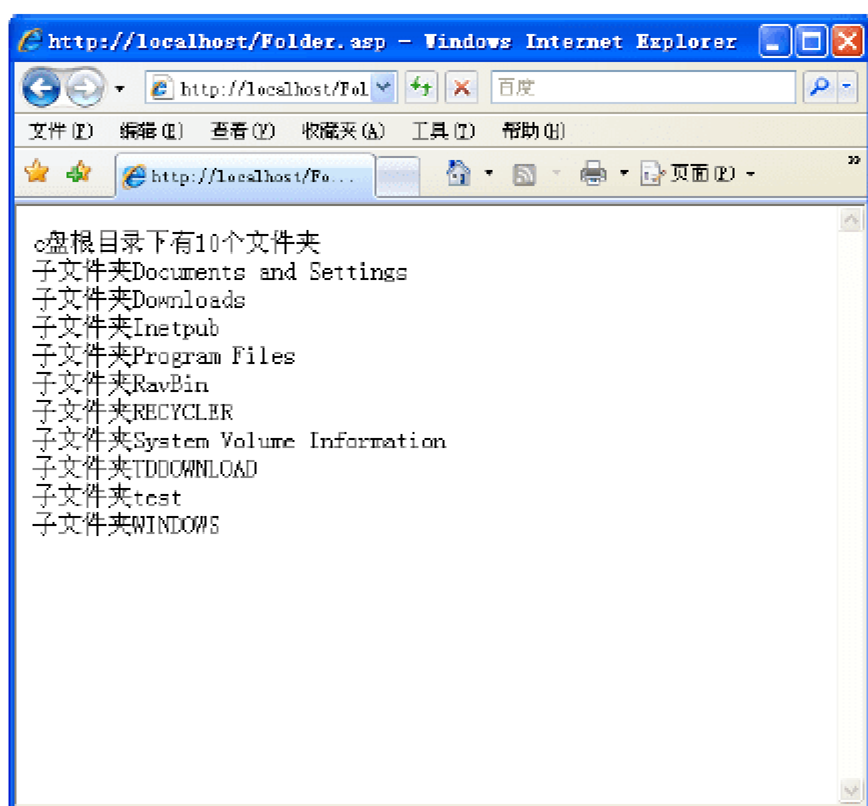


图 8-8　Folder.asp 的效果

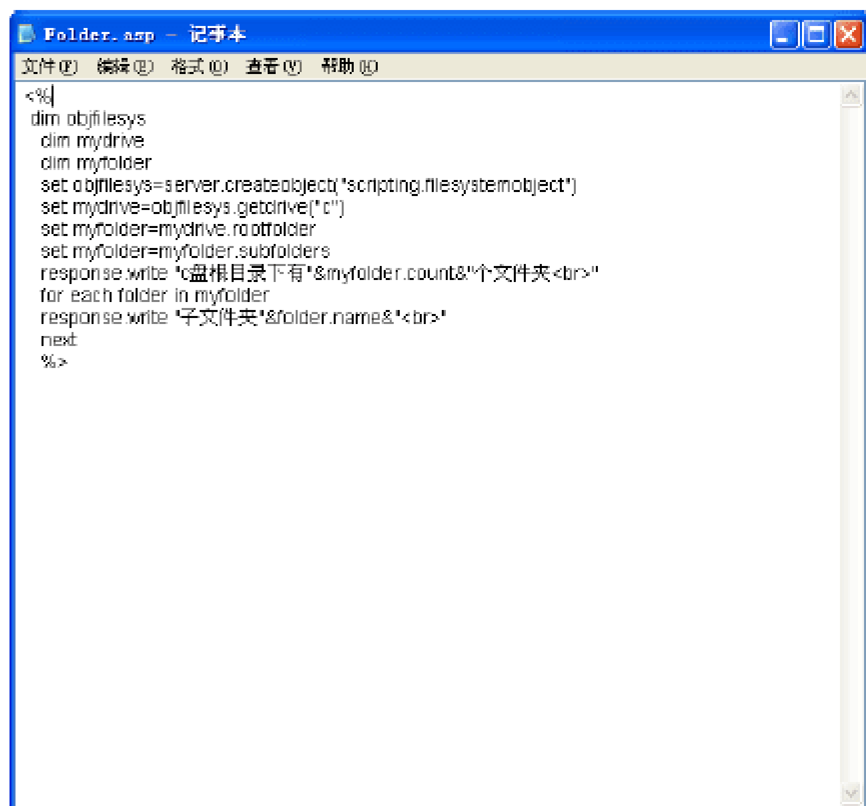


图 8-9　Folder.asp 的代码

【练习 8-8】创建一个文件，并显示这个文件的各属性值，程序运行效果如图 8-10 所示。

(1) 在 C:\Inetpub\wwwroot 文件夹中创建一个 test5.txt 文件。

(2) 创建一个名为 Folder2.asp 的文件，其代码如下所示：

```
<Title>检查文件属性</Title>
<Center><H2>查看文件的属性</H2></Center>
<Hr>
<%
whichfile=Server.MapPath("test5.txt")
Set fso = CreateObject("Scripting.FileSystemObject")
Set f1 = fso.CreateTextFile(whichfile,true)
f1.Write ("这是一个测试文件.")
f1.Close
Set f2 = fso.GetFile(whichfile)

s = "文件名称：" & f2.name & "<br>"
s = s & "文件短路径名：" & f2.shortPath & "<br>"
s = s & "文件物理地址：" & f2.Path & "<br>"
s = s & "文件属性：" & f2.Attributes & "<br>"
s = s & "文件大小：  " & f2.size & "<br>"
s = s & "文件类型：  " & f2.type & "<br>"
s = s & "文件创建时间：  " & f2.DateCreated & "<br>"
s = s & "最近访问时间: " & f2.DateLastAccessed & "<br>"
s = s & "最近修改时间: " & f2.DateLastModified
response.write(s)
%>
```

(3) 将 Folder2.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(4) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Folder2.asp 并按 Enter 键，效果如图 8-11 所示。



图 8-10　Folder.asp 的效果



图 8-11　Folder2.asp 的代码

注意：

在运行本节实例时，右击 C:\Inetpub\wwwroot 文件夹，在弹出的快捷菜单中选择“属性”命令，打开“属性”对话框。选择“安全”选项卡，并将 use 或 everyone 用户的权限全部选中。

【练习 8-9】用文本文件编写一个站点计数器，程序运行效果如图 8-12 所示。

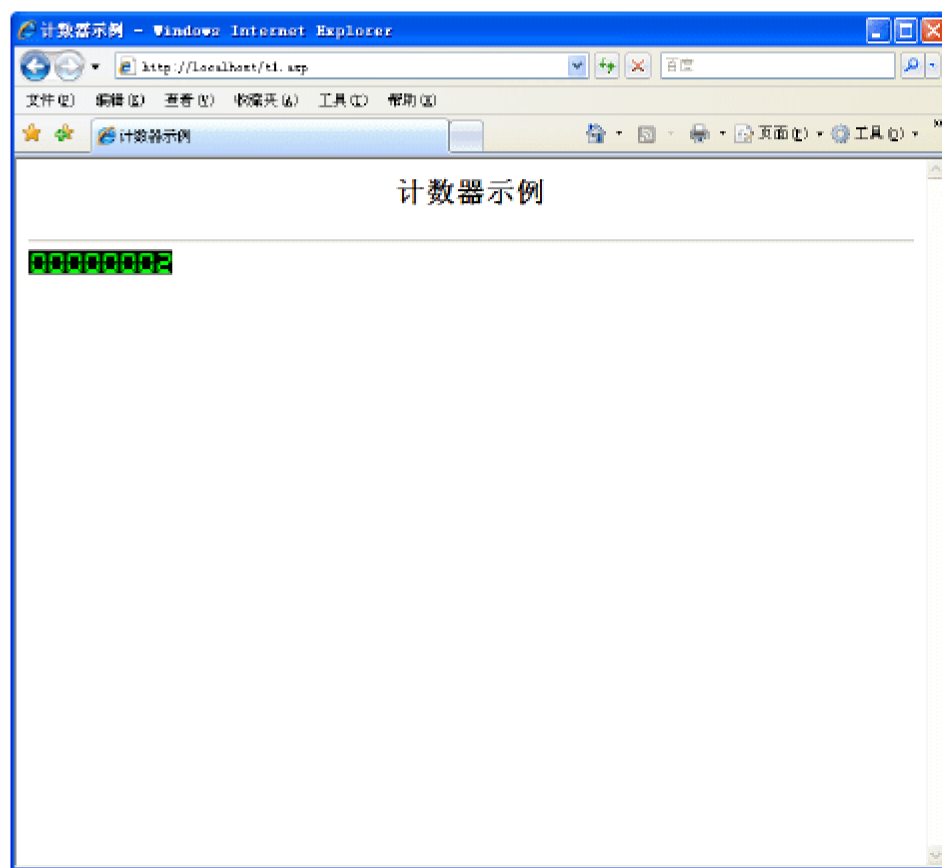


图 8-12　计数器效果

(1) 在 C:\Inetpub\wwwroot 文件夹中创建一个名为 num 的子文件夹，然后将如图 8-13 所示的图片文件复制至该文件夹内。

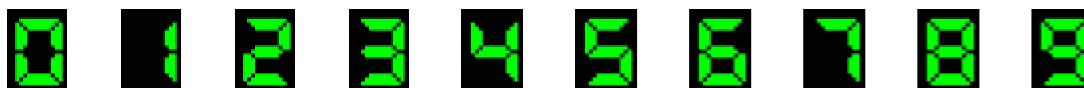


图 8-13　复制图片文件

(2) 创建一个名为 Folder3.asp 的文件，其代码如下(如图 8-14 所示):

```
<Title>计数器示例</Title>
<Center><H2>计数器示例</H2></Center>
<Hr>
<%
dim fso,f,i,counts,length
Const ForReading = 1
Const ForWriting = 2

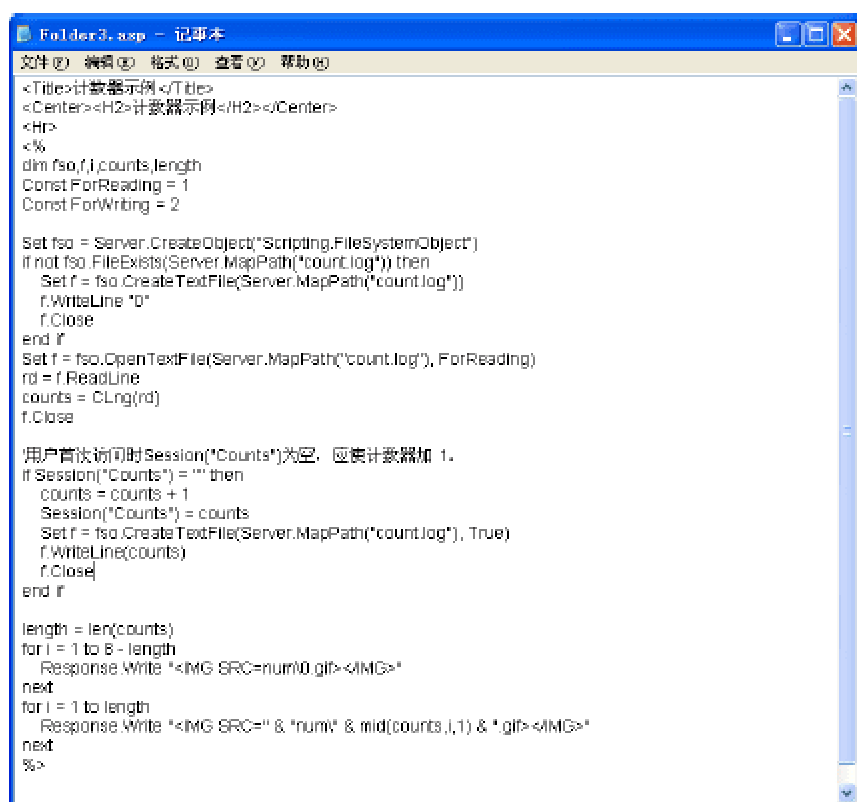
Set fso = Server.CreateObject("Scripting.FileSystemObject")
if not fso.FileExists(Server.MapPath("count.log")) then
    Set f = fso.CreateTextFile(Server.MapPath("count.log"))
    f.WriteLine "0"
    f.Close
end if
Set f = fso.OpenTextFile(Server.MapPath("count.log"), ForReading)
rd = f.ReadLine
counts = CLng(rd)
f.Close

'用户首次访问时，Session("Counts")为空，计数器加 1
if Session("Counts") = "" then
    counts = counts + 1
    Session("Counts") = counts
    Set f = fso.CreateTextFile(Server.MapPath("count.log"), True)
    f.WriteLine(counts)
    f.Close
end if

length = len(counts)
for i = 1 to 8 - length
    Response.Write "<IMG SRC=num\0.gif></IMG>"
next
for i = 1 to length
    Response.Write "<IMG SRC=" & "num\" & mid(counts,i,1) & ".gif></IMG>"
next
%>
```

(3) 将 Folder3.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(4) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Folder3.asp 并按 Enter 键，效果如图 8-12 所示。



```
<Title>计数器示例</Title>
<Center><H2>计数器示例</H2></Center>
<Hr>
<%
dim fso,f,i,counts,length
Const ForReading = 1
Const ForWriting = 2

Set fso = Server.CreateObject("Scripting.FileSystemObject")
if not fso.FileExists(Server.MapPath("count.log")) then
   Set f = fso.CreateTextFile(Server.MapPath("count.log"))
   f.WriteLine "0"
   f.Close
end if
Set f = fso.OpenTextFile(Server.MapPath("count.log"), ForReading)
rd = f.ReadLine
counts = CLng(rd)
f.Close

'用户首次访问时Session("Counts")为空，应使计数器加 1。
if Session("Counts") = "" then
   counts = counts + 1
   Session("Counts") = counts
   Set f = fso.CreateTextFile(Server.MapPath("count.log"), True)
   f.WriteLine(counts)
   f.Close
end if

length = len(counts)
for i = 1 to 8 - length
   Response.Write "<IMG SRC=num\0.gif></IMG>"
next
for i = 1 to length
   Response.Write "<IMG SRC=" & "num\" & mid(counts,i,1) & ".gif></IMG>"
next
%>
```

图 8-14　Folder3.asp 的代码

8.4　获得驱动器的信息

通过 Drives 对象提供的属性，可以获得系统上各个驱动器有关的信息，这些驱动器可以是物理的或通过网络逻辑连接到系统上的。Drives 对象的属性及其相关说明如表 8-7 所示。

表 8-7　Drive 对象的属性及说明

属　性	说　明
AvailableSpace	返回驱动器对于该用户可用的空间大小
DriveLetter	返回驱动器的字母
DriveType	返回驱动器的类型。如可移动的、固定的、网络、CD-ROM 或 RAM 盘
FileSystem	返回驱动器文件系统的类型。返回值包括 FAT 和 NTFS 等
FreeSpace	返回驱动器可用剩余空间的总量
IsReady	返回一个布尔值，表明驱动器是否已准备好
Path	返回一个由驱动器字母和冒号组成的驱动器路径，如 C:
RootFolder	返回代表驱动器根目录文件夹的 Folder 对象
SerialNumber	返回一个用于识别磁盘卷的十进制的序列号
ShareName	如果是一个网络驱动器，返回该驱动器的网络共享名
TotalSize	返回驱动器的总容量(以字节为单位)
VolumeName	设定或返回本地驱动器卷名

【练习 8-10】使用 Drives 对象显示系统上所有驱动器的信息，程序运行效果如图 8-15 所示。

(1) 创建一个名为 Drives.asp 的文件，其代码如下所示(如图 8-16 所示):

```
<title>检测驱动器</Title>
<Center><H2>您的计算机上各驱动器相关信息</H2>
<Hr>
<%
Function tran(Driver)
Select Case Driver
Case 0: tran="设备无法识别"
Case 1: tran="软盘驱动器"
Case 2: tran="硬盘驱动器"
Case 3: tran="网络硬盘驱动器"
Case 4: tran="光盘驱动器"
Case 5: tran="RAM 虚拟磁盘"
End Select
End Function
set fso=Server.CreateObject("Scripting.FileSystemObject")
%>
<table border=1 width="100%">
<tr>
<td>盘符</td>
<td>类型</td>
<td>卷标</td>
<td>总计大小</td>
<td>可用空间</td>
<td>文件系统</td>
<td>序列号</td>
<td>是否可用</td>
<td>路径</td>
</tr>
<%
'如果系统的某个驱动器中没有磁盘，例如软驱中无软盘或 CD-ROM 驱动器里没有光盘，
'将得到一个错误提示：“驱动器不存在”
'使用错误处理语句可强行将程序跳转去检查下一驱动器
on error resume next
For each drv in fso.Drives
    Response.Write "<tr>"
    Response.Write "<td>" & drv.DriveLetter & "</td>"
    Response.write "<td>" & tran(drv.DriveType) & "</td>"
    Response.write "<td>" & drv.VolumeName & "</td>"
    Response.write "<td>" & FormatNumber(drv.TotalSize / 1024, 0)& "</td>"
    Response.write "<td>" & FormatNumber(drv.Availablespace / 1024, 0) & "</td>"
    Response.write "<td>" & drv.FileSystem & "</td>"
    Response.write "<td>" & drv.SerialNumber & "</td>"
    Response.write "<td>" & drv.IsReady & "</td>"
```

```
        Response.write "<td>" & drv.Path & "</td>"
        Response.Write "</tr>"
    Next
    set fs=nothing
    %>
    </table></Center>
```

(2) 将 Drives.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Drives.asp 并按 Enter 键，效果如图 8-15 所示。



图 8-15　Drives.asp 的效果



图 8-16　Drives.asp 的代码

8.5　对文件夹进行操作

通过 Folder 对象提供的属性和方法，可对文件夹进行各种操作，其中包括提取文件夹信息，创建、删除、复制和移动文件夹等。

8.5.1　Folder 对象的属性

Folder 对象提供一组属性，可用这些属性得到关于当前文件夹的更多信息，也可以改变该文件夹的名称。Folder 对象的属性及其相关说明如表 8-8 所示。

表 8-8　Folder 对象的属性及说明

属　　性	说　　明
Attributes	返回文件夹的属性。可以是下列值中的一个或其组合，正常为 0，只读为 1，隐藏为 2，系统为 4，卷为 8，文件夹为 16，存档为 32，别名为 64，压缩为 128。例如，一个隐藏的只读文件，Attributes 的值为 2

(续表)

属　　性	说　　明
DateCreated	返回该文件夹的创建日期和时间
DateLastAccessed	返回最后一次访问该文件夹的日期和时间
DateLastModified	返回最后一次修改该文件夹的日期和时间
Drive	返回该文件夹所在的驱动器字母
Files	返回 Folder 对象包含的 Files 集合，表示该文件夹内所有的文件
IsRootFolder	返回一个布尔值，说明该文件夹是否是当前驱动器的根文件夹
Name	设定或返回文件夹的名字
ParentFolder	返回该文件夹的父文件夹对应的 Folder 对象
Path	返回文件夹的绝对路径，使用相应的长文件名
ShortName	返回按照早期 8.3 文件命名约定转换的短文件名
ShortPath	返回按照早期 8.3 文件命名约定转换的短文件绝对路径
Size	返回包含在该文件夹中所有文件和子文件夹的大小
SubFolders	返回该文件夹内包含的所有子文件夹对应的 Folders 集合，包括隐藏文件夹和系统文件夹
Type	如果可能，返回一个文件夹类型的说明字符串，例如，RecycleBin

8.5.2　Folder 对象的方法

Folder 对象提供一组可用于复制、删除和移动当前文件夹的方法。这些方法的运行方式与 FileSystemObject 对象的 CopyFolder、DeleteFolder 和 MoveFolder 方法相同。但 Folder 对象的方法不要求 source 参数，因为源文件就是这个文件夹。Folder 对象的方法及其相关说明如表 8-9 所示。

表 8-9　Folder 对象的方法及说明

方　　法	说　　明
Copy(destination，overwrite)	将这个文件夹及里面所有的内容复制到 destination 指定的文件夹
Delete(force)	删除文件夹及里面的所有内容
Move(destination)	将文件夹及里面所有的内容移动到 destination 指定的文件夹
CreateTextFile (filename,overwrite,unicode)	用指定的文件名在文件夹内创建一个新的文本文件，并且返回一个相应的 TextStream 对象

【练习 8-11】使用 Folder 对象执行建立和删除文件夹操作，程序运行效果如图 8-17 所示。

(1) 创建一个名为 Folder4.asp 的文件，其代码如下所示：

```
<%@ Language=VBScript %>
<%
Sub CreateAFolder(file)
   Dim fso
   Set fso = CreateObject("Scripting.FileSystemObject")
   fso.CreateFolder(file)
   response.write "已经建立了"&file
End Sub
Sub DeleteAFolder(file)
   Dim fso
   Set fso = CreateObject("Scripting.FileSystemObject")
   fso.DeleteFolder(file)
   response.write "已经删除了"&file
End Sub
%>
<%
subname=request.form("submit")
create=request.form("create")
del=request.form("del")
if subname<>"" then
   if create<>"" then
      call CreateAFolder(""&create&"")
   end if
   if del<>"" then
      call DeleteAFolder(""&del&"")
   end if
end if
%>
<form action="<%=Request.ServerVariables("Script_Name")%>" method="post">
<input name="create">
<input type="submit" value="建立一个文件夹" name="submit">
</form>
<hr>
<form action="<%=Request.ServerVariables("Script_Name")%>" method="post">
<input name="del">
<input type="submit" value="删除文件夹" name="submit">
</form>
```

(2) 将 Folder4.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Folder4.asp 并按 Enter 键，效果如图 8-17 所示。

(4) 在 Folder4.asp 页面的第一个对话框中输入 C:\Inetpub\wwwroot\test 后，单击“建立一个文件夹”按钮，程序将自动在 C:\Inetpub\wwwroot 文件夹中创建一个子文件夹 test，如

图 8-18 所示。

(5) 在 Folder4.asp 页面的第二个对话框中输入 C:\Inetpub\wwwroot\test 后，单击“删除文件夹”按钮，程序将会把 C:\Inetpub\wwwroot 文件夹中创建的 test 文件夹删除。

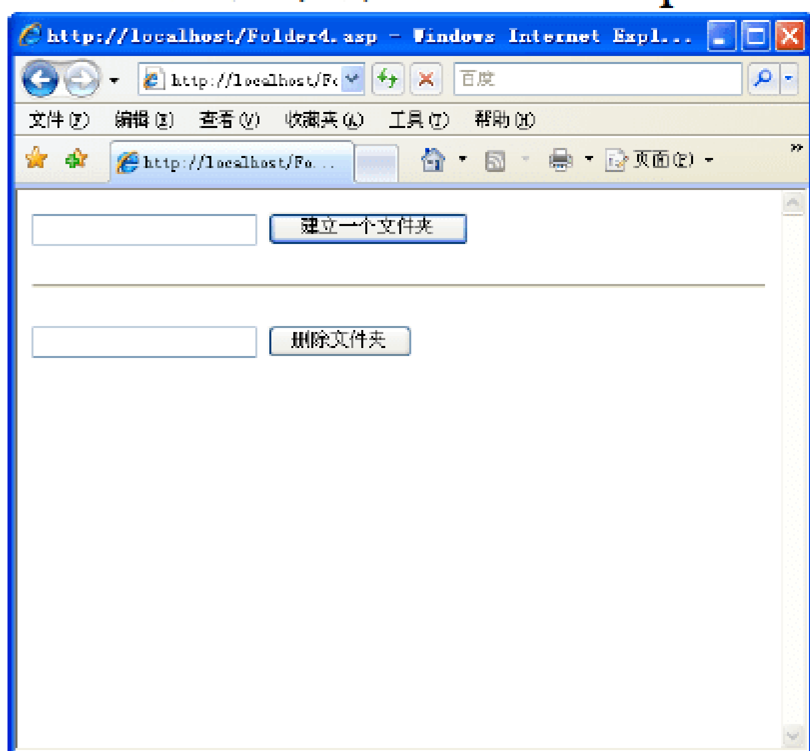


图 8-17　Folder4.asp 的效果

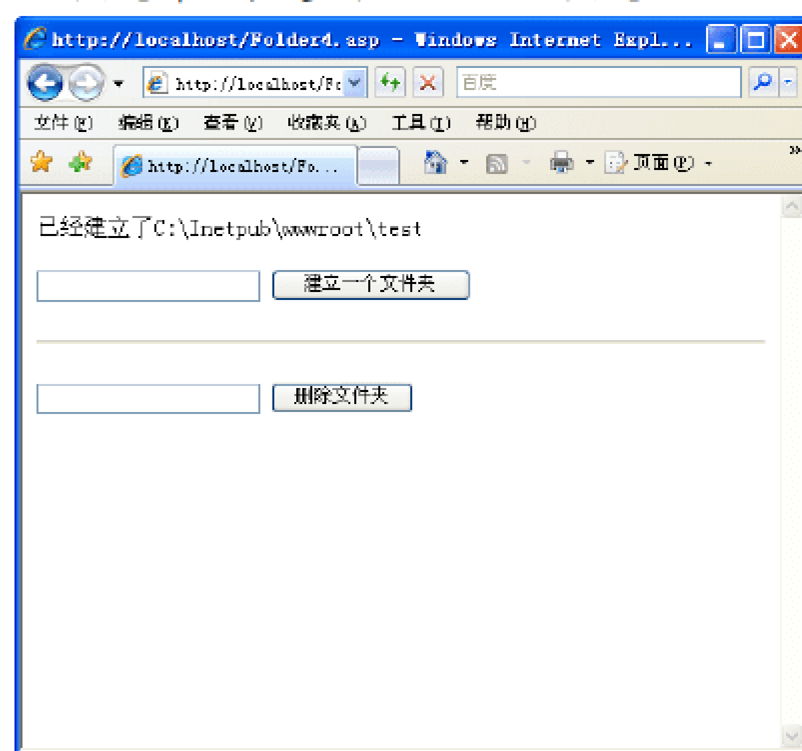


图 8-18　创建文件夹

8.6　习　　题

8.6.1　填空题

(1) FSO 组件可以用来对__________、__________以及文件等对象进行相关的处理操作。

(2) 使用 FSO 编程，首先要用____________方法创建 FileSystemObject 对象。

(3) TextStream 对象的___________方法用来写入一整行数据至已打开的数据流文件。

8.6.2　选择题

(1) 下面不能创建或打开一个文本文件并返回 TextStream 对象的方法是(　　)。

A. CreateTextFile 方法

B. WriteBlankLines 方法

C. OpenAsTextStream 方法

D. OpenTextFile 方法

(2) 下面可打开文件的方法有(　　)。

A. File 对象的 OpenAsTextStream 方法

B. FileSystemObject 对象的 OpenAsTextStream 方法

C. File 对象的 OpenTextFile 方法

D. TextStram 对象的 OpenAsTextStream 方法

(3) 下面不能从文本文件中读取数据的有(　　)。

A. TextStream 对象的 Read 方法

B. TextStream 对象的 ReadLine 方法

C. TextStream 对象的 ReadAll 方法

D. TextStream 对象的 OpenAsTextStream 方法

8.6.3　问答题

(1) 什么是 File Access 组件的 FSO 对象？

(2) 如何打开文本文件并添加数据？

8.6.4　操作题

(1) 列表显示站点根目录中的文件夹和文件，程序运行效果如图 8-19 所示。

(2) 编制一个语句接龙页面，程序运行效果如图 8-20 所示。

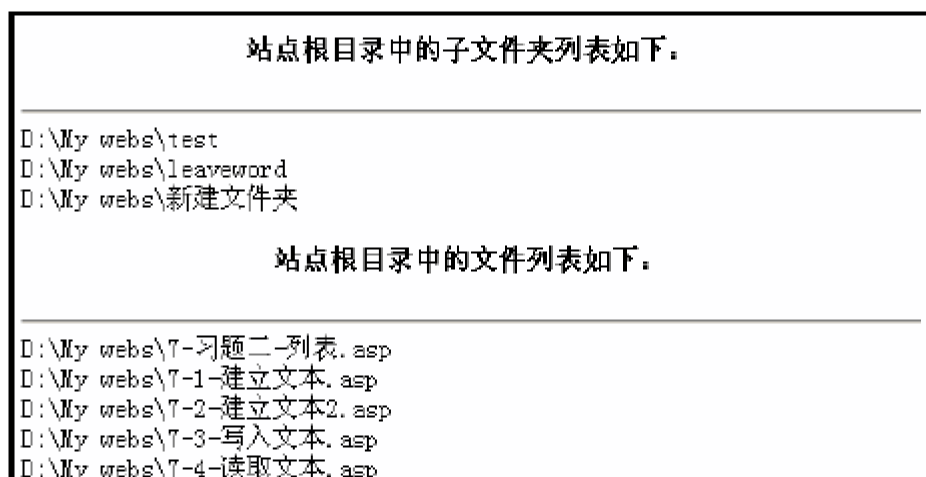


图 8-19　网页效果 1

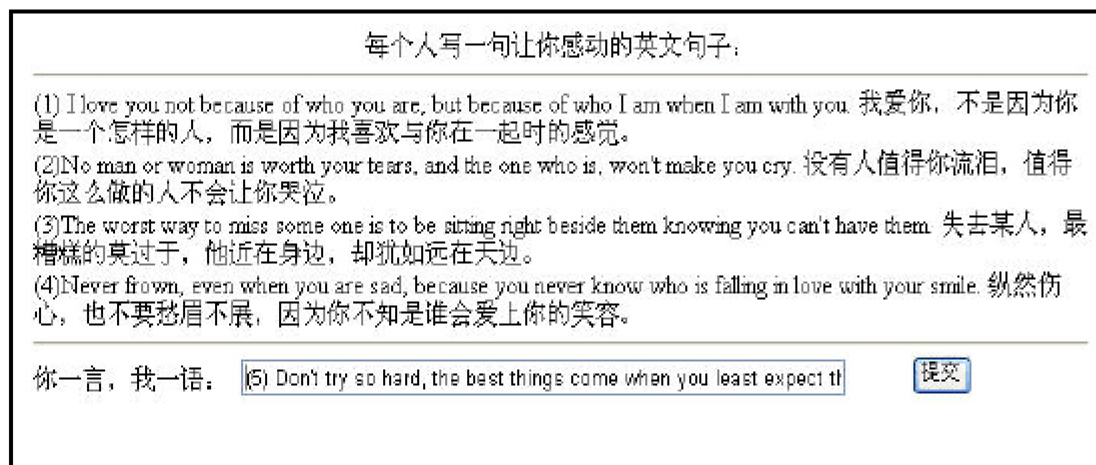


图 8-20　网页效果 2

第9章　利用ADO组件访问数据库

数据库是管理大量、一致、可靠、共享、持久的数据资源的计算机软件产品。数据库软件是动态网站实现互动效果的一个非常重要的核心组件，是支撑以电子商务、网上论坛和信息发布为代表的一系列网络服务的坚强支柱。在动态网页的开发工作中，应用 ASP 内建的 Database Access 组件，可以非常方便地通过 ActiveX Date Objects(ADO)对象访问存储在服务器端的数据库中的信息。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 了解数据库的概念和常用术语
- 了解 ADO 数据模型
- 掌握创建 Connection 对象的方法
- 掌握 Command 对象的基本用法
- 掌握创建记录集的方法

9.1　应用 Access 数据库

数据库的主要功能是存储与管理数据，在目前市面上存在的多种形式的数据库产品中，关系型数据库最受欢迎并被广泛使用。该类数据库将数据按类别存储在各种数据表中，并且通过数据表之间的关联进行数据的调整和搜索等维护操作。

9.1.1　快速掌握 Microsoft Access

Access 数据库是目前比较流行的数据库管理系统，它是一个运行在 Windows 系统环境下的桌面关系型数据库，也是 Office XP 的组件之一。下面将围绕几个与创建 Access 数据库相关的问题进行讲述，以便用户对该数据库的应用有个概括的了解。

1. Access 数据库简介

Access 数据库属于比较简单的小型数据库系统，一般用于小型公司的数据管理。其基本数据库观念与其他大型数据库系统(如 Oracle 和 SQL Server 等)并没有太大的差别。

在启动 Microsoft Access 2003 后，用户首先看到的是版权信息，选择“文件”|“新建”命令创建一个库文件后，就可以进入如图 9-1 所示的工作界面。Access 2003 的工作界面包括菜单栏、工具栏、工作区和状态栏等几个部分。

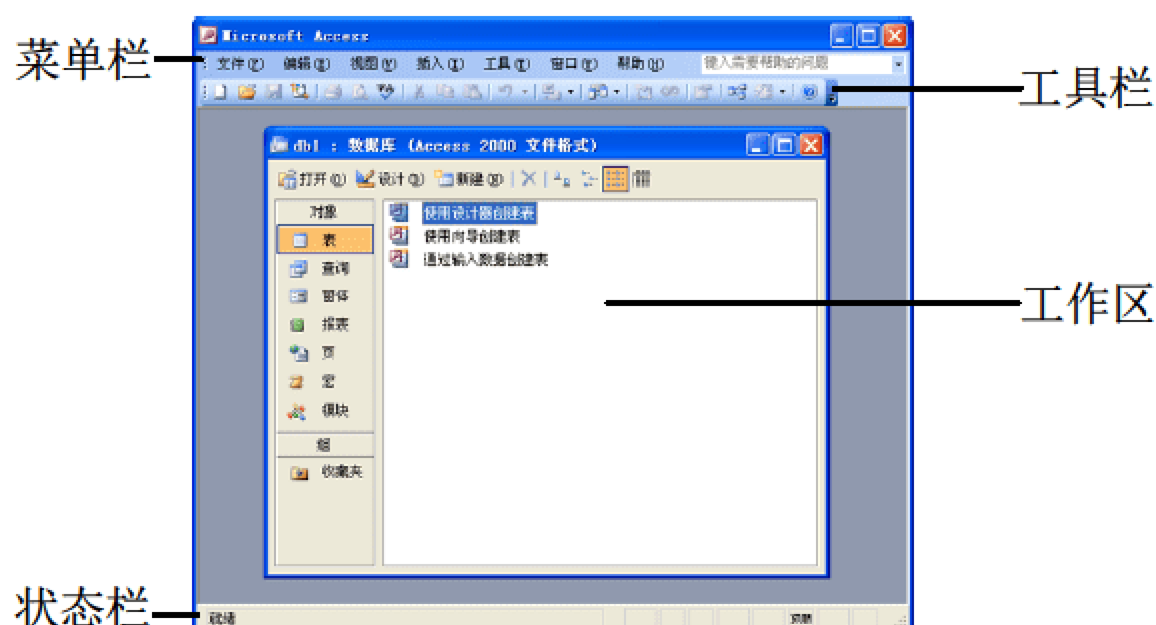


图 9-1　Microsoft Access 2003 的工作界面

2. 创建数据库

Access 数据库将数据按类别存储在不同的数据表中，以方便数据的管理和维护。用户要设计数据表，首先要创建一个数据库。下面将以一个简单的实例，详细介绍建立 Access 2003 数据库的操作步骤。

【练习 9-1】在 C:\Inetpub\wwwroot 目录中新建一个名为 AcDatabase 的目录，然后利用 Access 2003 数据库软件创建一个名为 db1 客户信息数据库，并将其保存在新建的目录中。

(1) 打开 C:\Inetpub\wwwroot 目录，选择“文件”|“新建”|“文件夹”命令，创建一个名为 AcDatabase 的子目录。

(2) 选择“开始”|“程序”| Microsoft Office| Access 2003 命令，启动 Access 2003 数据库，然后选择“文件”|“新建”命令，打开“新建文件”对话框，如图 9-2 所示。

(3) 单击“空数据库”链接，在打开的“文件新建数据库”对话框的“保存位置”下拉列表框中选择步骤(1)所创建的 AcDatabase 目录，在“文件名”文本框中输入 db1(如图 9-3 所示)，然后单击“创建”按钮。

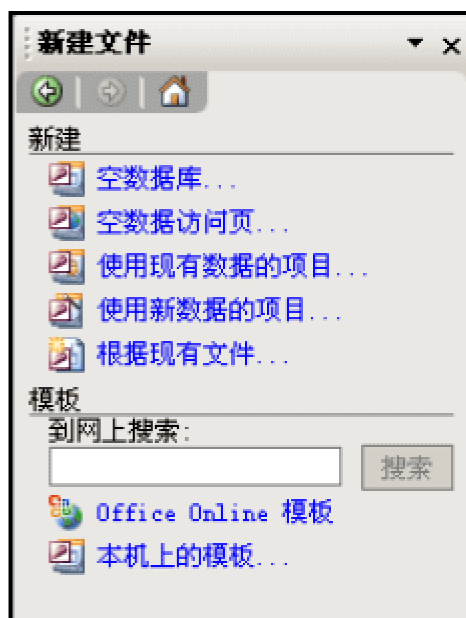


图 9-2　“新建文件”对话框

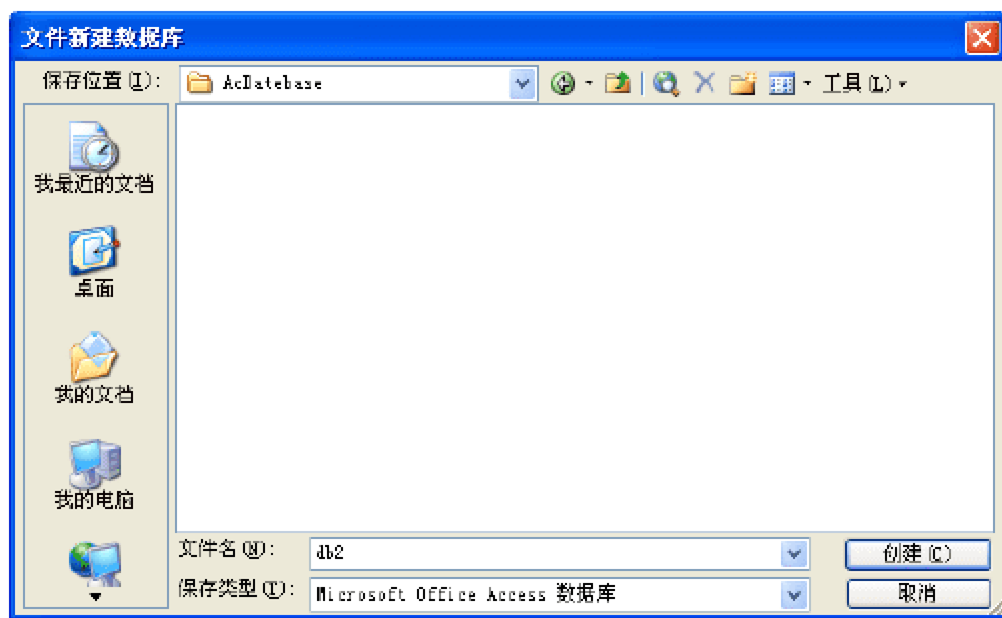


图 9-3　“文件新建数据库”对话框

(4) 在打开的 Access 2003 工作界面的工作区域中选择“对象”列表框中的“表”选项，然后在对话框右侧的选项区域中双击“使用设计器创建表”选项(如图 9-4 所示)，打开数据表“表 1”的设计检视窗口。

(5) 设置数据表中字段的结构。首先在“表 1”的设计检视窗口的“字段名称”列的第一个单元格中输入 C-Id，在其后的“数据类型”下拉列表中选择“自动编号”选项(如图 9-5 所示)，在“说明”列的第一个单元格中输入对表格字段的描述文本。

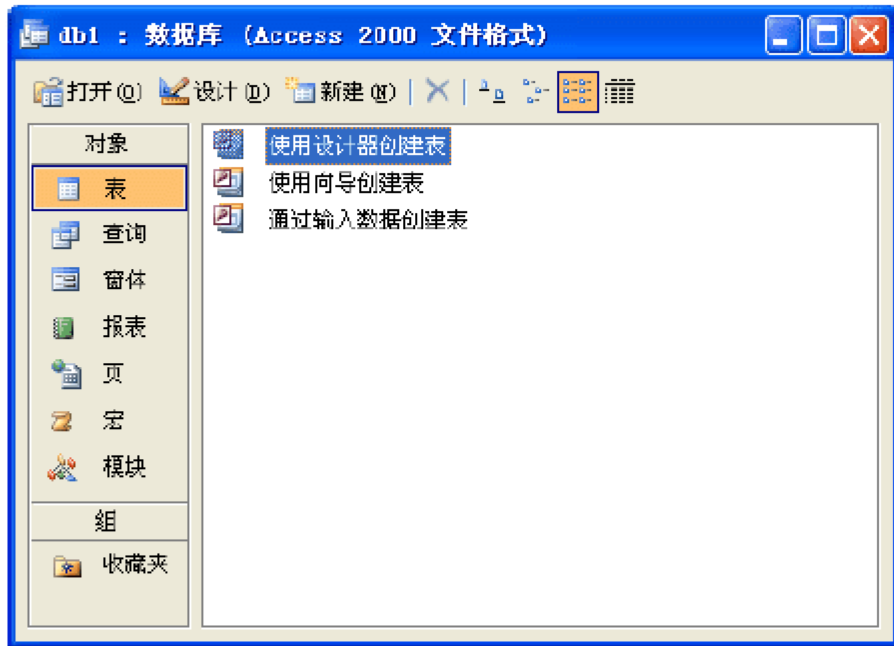


图 9-4　设置数据库

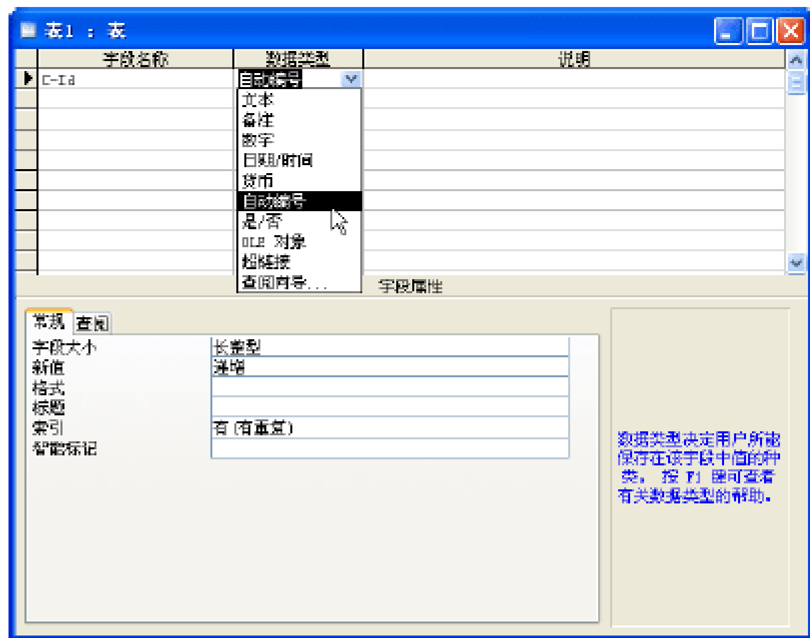


图 9-5　设置数据表字段结构

(6) 右击 C-Id 字段，在弹出的快捷菜单中选择“主键”命令，为 C-Id 字段前添加🔑标志，将该字段设置为主键，如图 9-6 所示。

(7) 在“表 1”的设计检视窗口第 2 行的“字段名称”中输入 C-Name，在“数据类型”下拉列表中选择“文本”，在“说明”中输入对表格字段的描述文本。

(8) 选中 C-Name 字段，在下面的“字段属性”选项区域中选择“常规”选项卡。在“字段大小”文本框中输入 30，在“必填字段”下拉列表中选择“是”选项，在“允许空字符串”下拉列表中选择“否”选项，如图 9-7 所示。

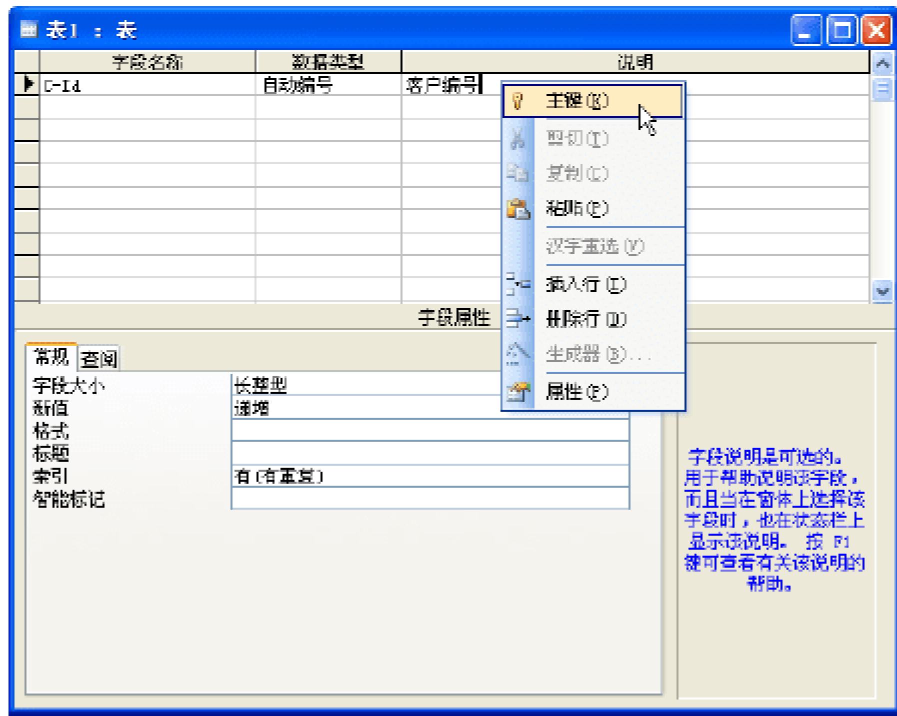


图 9-6　设置数据表主键

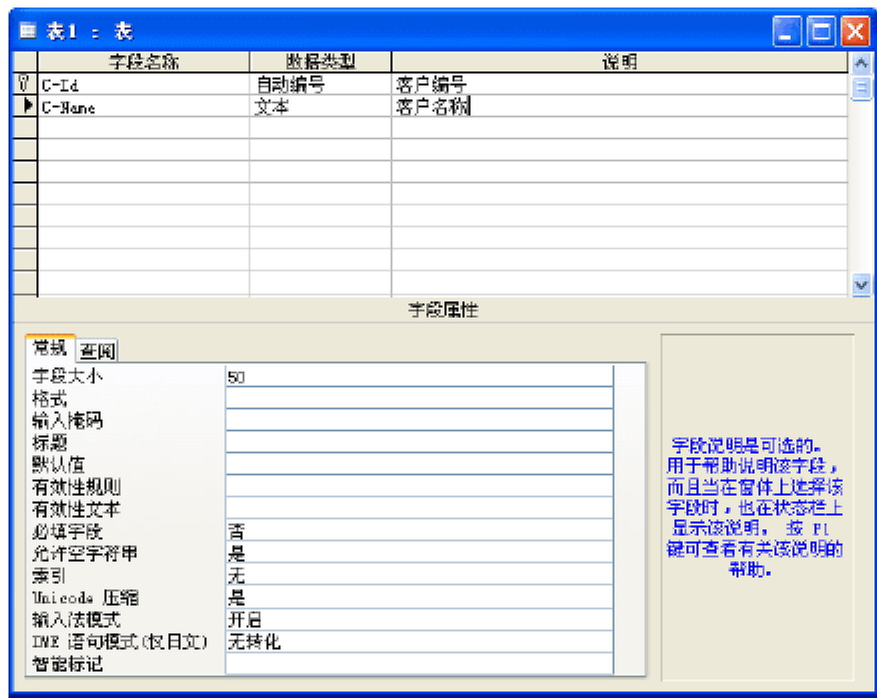


图 9-7　设置 C-Name 字段参数

(9) 重复步骤(6)和(7)的操作，使用相同的方法将如表 9-1 所示的内容输入到“表 1”的设计检视窗口中。

表 9-1　创建“表 1”字段参照表

字段名称	数据类型	说　明	字段大小	必填字段	是否允许空字符串	备　注
C-Id	自动编号	客户编号	—	—	—	主键
C-Name	文本	客户名称	30	是	否	—
C-address	文本	客户地址	100	否	是	—
C-Tel	文本	客户电话	30	是	是	—
C-Mail	文本	客户 E-Mail	20	否	是	—

(10) 完成字段录入工作后，“表 1”的设计检视窗口如图 9-8 所示。选择“文件”|“保存”命令将数据表保存。这时，在 Access 的工作区域中将会看到数据表“表 1”已经自动添加在列表框中，如图 9-9 所示。

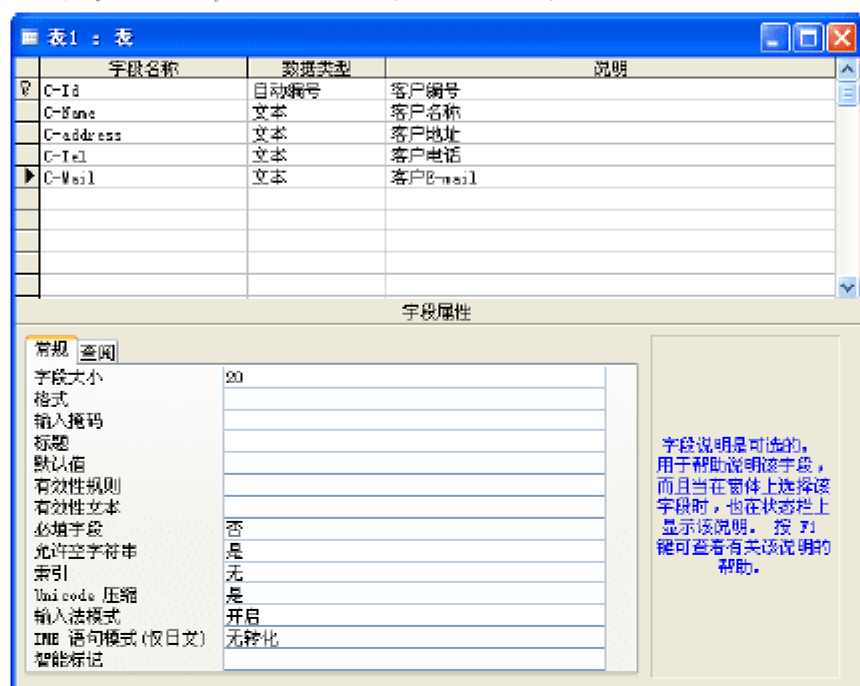


图 9-8　完成字段创建后的效果

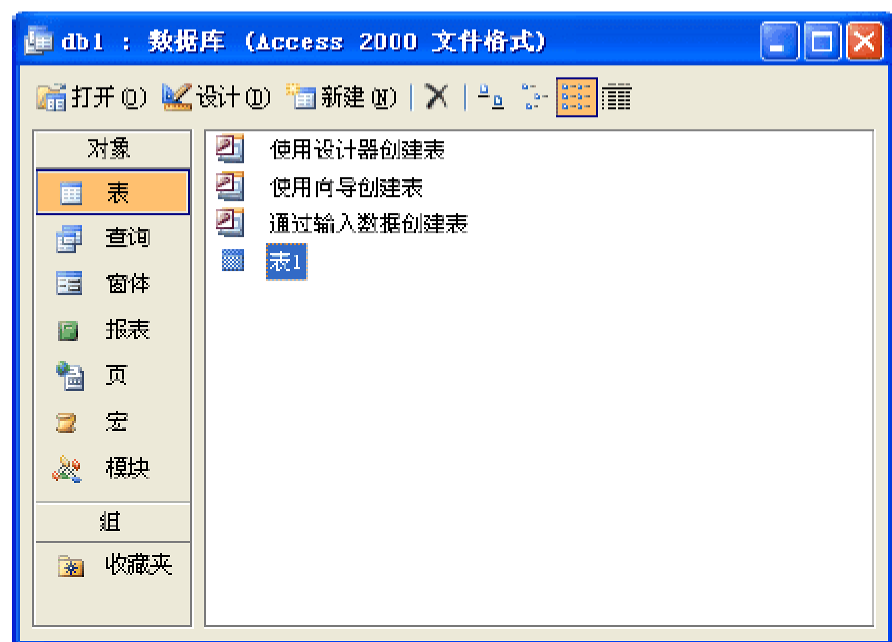


图 9-9　完成数据库的创建

注意：

Access 数据库操作非常简单，该数据库所创建的库文件是一种扩展名为 mbd 的特殊文件，如果用户的计算机上已经安装了 Access 数据库软件，双击这种文件就可以直接打开 Access 数据库的编辑窗口。

9.1.2　Access 数据库基础

本节将以【练习 9-1】创建的数据库 db1 为基础，介绍组成 Access 数据库的结构及功能，包括数据库结构、数据表结构、数据内容和字段索引等内容。

1. 数据库结构

Access 数据库通常包含多个数据表。以【练习 9-1】创建的数据库为例，用户可以按照同样的操作步骤，在数据库 db1 中创建“表 2”，“表 3”，“表 4”或任意名称的数据表(如图 9-10 所示)，但这些表都属于 db1。

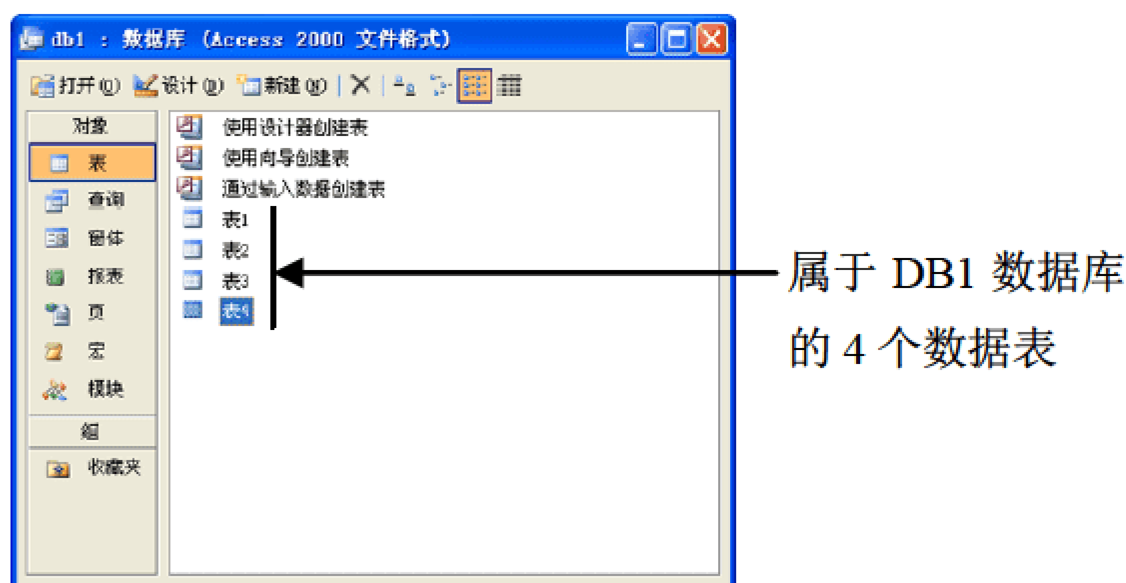


图 9-10　数据库的结构

注意：

关系型数据库通常包含多个内容不同的数据表，并通过数据表之间的特定字段定义其各自之间的关系。通过这种关系，可以在不同数据表中取得相关的数据内容。

2. 数据表结构

在数据表中包含两个重要的属性：字段名称和数据类型。字段名的作用是在数据表中识别字段；而数据类型则是该字段所能存储的数据类型，例如文字、数字和日期时间等。用户可以在数据库工作区中选中某个数据表，然后单击“设计”按钮设计(D)，打开表的设计检视窗口，对数据表的属性进行调整。

如图 9-11 所示就是【练习 9-1】所创建的数据表“表 1”的设计检视窗口。其中被选中的部分代表数据表“表 1”中“字段名称”为 C-Name 的字段，而窗口下面的“常规”选项卡包含了该字段的各种特性设置，例如，在“字段大小”文本框中设置字段所能存储的数据长度，在“必填字段”下拉列表中设置字段是否为必有值，在“允许空字符串”下拉列表中设置字段内容是否可为空字符串等。

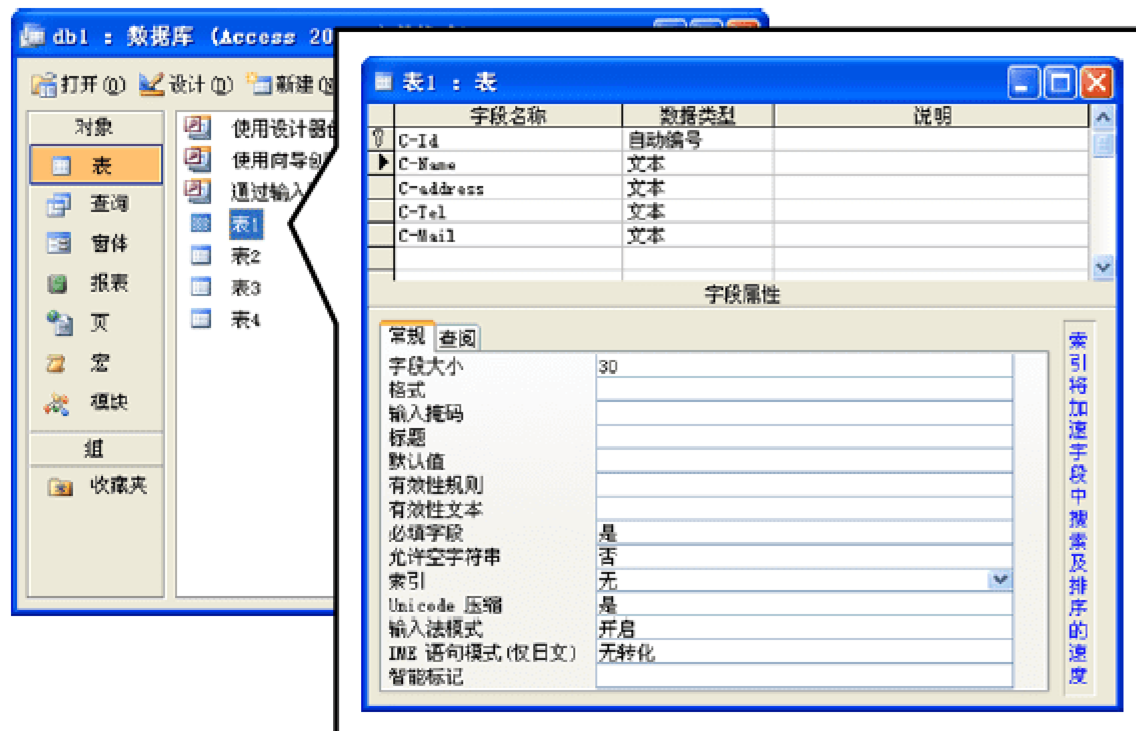


图 9-11　数据表“表 1”的设计检视窗口

另外，在数据表的设计检视窗口中，单击“数据类型”列单元格中的下拉列表按钮，在弹出的下拉列表中还可以为字段设置数据类型限制。以图 9-11 所显示的 C-Name 字段为例，它的数据类型为“文本”，如果要修改这项设置，可以单击该字段的“数据类型”下拉列表按钮，然后在弹出的下拉列表中选择所需的选项即可。

3. 数据内容

数据库中的数据内容存储于数据表中，在数据库工作区中双击数据表名称即可打开数据表。从外观上看，数据表类似 Excel 表格，每一行代表一个记录，每一列代表一个字段，并且每个字段都有其特定的字段名称和字段数据类型。图 9-12 所示就是在对【练习 9-1】创建的数据表“表 1”完成内容填充后的客户数据表。

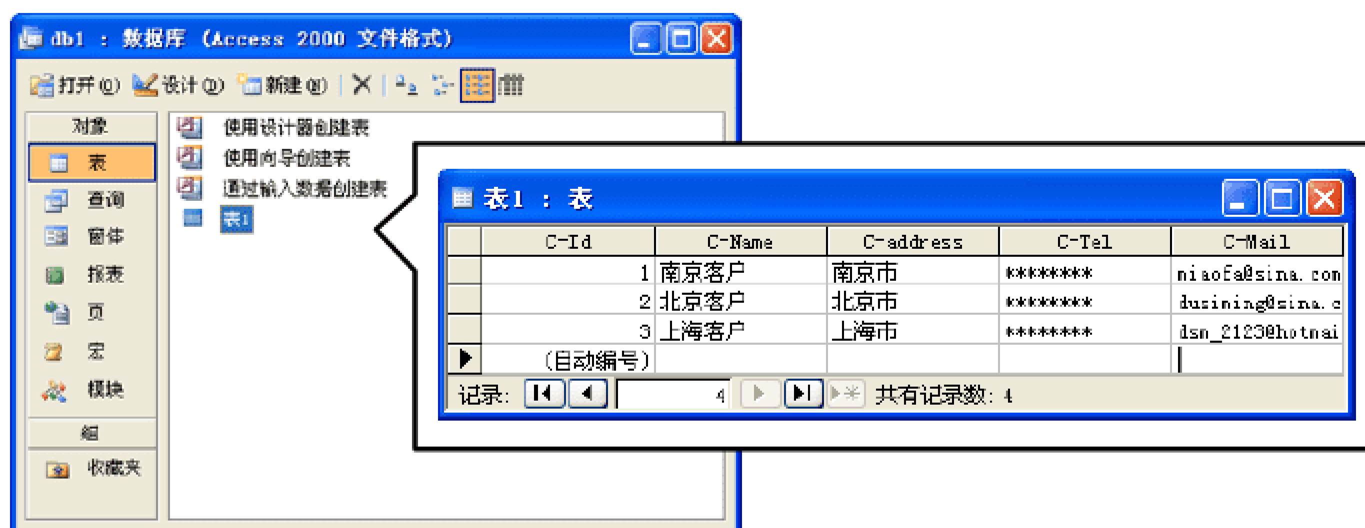


图 9-12　数据表“表 1”中的数据内容

注意：

单击数据表中的某项记录，就可以针对该记录进行修改，或直接按 Delete 键将其删除。

4. 字段索引

索引是一种字段标识，通常完成一个数据表字段设计后，还需要针对其中的字段设置索引。索引的主要功能有两种，增加数据的搜索速度和设置数据表关联，数据表的索引的作用就如同书签一样，数据库可以根据索引快速地查找到存储于数据表中的特定数据。而数据表之间的索引字段关联则可以串联不同数据表中的数据内容。

索引本身根据其功用可以分为两种：主键(主索引)和一般性索引。一个数据表只能有一个字段被设置为主键，而被设置为主键的字段在整个数据表中的数据内容是唯一值，不允许被重复。例如【练习 9-1】的步骤(5)将 C-Id 字段设置为主键，该字段作为数据表的客户编号字段，当用户在数据表中输入数据内容时将不会有重复的客户编号存在，这样数据库系统就可以根据这个字段的编号取得特定客户的数据内容。

一个数据表可以有多个字段被设置为一般性索引。这种索引的功用除了与其他数据表的主键字段关联以外，还可以加速数据库的搜索速度。

【练习 9-2】以【练习 9-1】所创建的数据库 db1 中的数据表“表 1”为基础，设置该数据表中字段 C-Id 与字段 C-Tel 之间的一般性索引。

(1) 在 Access 2003 中打开数据库 db1，然后选中数据表“表 1”，并单击“设计”按钮，打开设计检视窗口。

(2) 在数据表“表 1”的设计检视窗口中选中字段 C-Id，然后在工具栏中单击“索引”按钮(如图 9-13 所示)，打开“索引：表 1”对话框。

(3) 在“索引名称”文本框中输入 C-Id，然后单击该文本框后的“字段名称”下拉列表按钮，在弹出的下拉列表中选择 C-Tel 选项(如图 9-14 所示)。

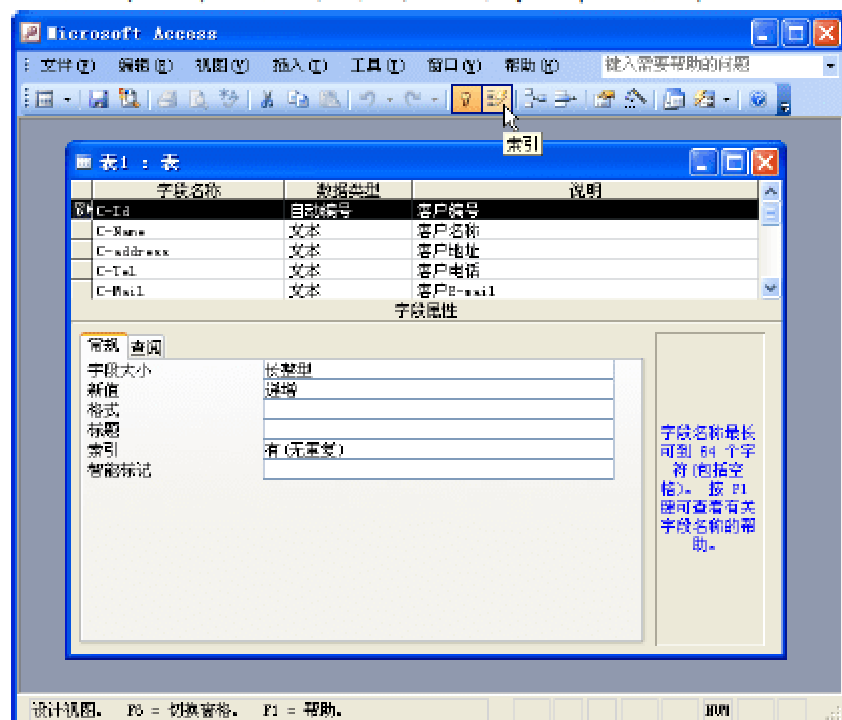


图 9-13　设置索引



图 9-14　“索引：表 1”对话框

(4) 完成设置后，关闭“索引：表 1”对话框，在数据表“表 1”的 C-Tel 字段里成功新增一个名为 C-Id 的字段索引。

9.1.3　结构化查询语言简介

SQL 是结构化查询语言(Structured Query Language)的缩写，包括查询、定义、操纵和控制 4 个部分，是一种功能齐全的数据库语言。

数据查询是指按要求查找出满足条件的记录的操作。数据定义是指对关系模式一级的定义。数据操纵是指对关系中的具体数据进行增、删、改和更新等操作。数据控制是指对数据访问权限的授予或撤销。

SQL 具有语言简洁、方便实用、功能齐全等优点。目前，多数数据库管理系统都支持 SQL 或提供 SQL 接口。

1. SQL 的构成

SQL 是由命令、子句和运算符等元素所构成的，这些元素结合起来组成用于创建、更新和操作数据库的语句。SQL 语言常分为 3 类；数据操纵语言 DML、数据定义语言 DDL 和数据库控制语言 DCL。

数据定义语言 DDL 完成定义数据库的结构，包括数据库本身、数据表、目录、视图等数据库元素。常用 DDL 语句及其相关说明如表 9-2 所示。

表 9-2　常用 DDL 语句及其相关说明

常用 DDL 语句	说　　明
CREATE TABLE	创建表
CREATE INDEX	创建索引
CREATE VIEW	创建视图

(续表)

常用 DDL 语句	说　明
ALTER TABLE	增加表列，重定义表列，更改存储分配
DROP TABLE	删除表
DROP INDEX	删除索引

数据操纵语言 DML 完成在数据库中确定、修改、添加、删除某一数据值的任务。常用 DML 语句及其相关说明如表 9-3 所示。

表 9-3　常用 DML 语句及其相关说明

常用 DML 语句	说　明
SELCECT	在数据库中查找满足指定条件的记录
INSERT	增加数据行到表
DELETE	从表中删除数据行
UPDATE	更改表中的数据

数据库控制语言 DCL 用来授予或回收访问数据库的某种特权，并控制数据库操纵事务发生的时间及效果，对数据库实行监视等。常用 DCL 语句及其相关说明如表 9-4 所示。

表 9-4　常用 DCL 语句及其相关说明

DCL 语句	说　明
GRANT	将权限或角色授予用户或其他角色
REVODE	回收用户权限
ROLL	回滚。这是一个把信息恢复到使用 UPDATE、INSERT、DELETE 前最后提交的状态
COMMIT	提交。在完成数据库的插入、删除和修改操作时，只有当事务提交到数据库才算完成

一个 SQL 语句是一个由从句组成的命令，从句指定了要执行的操作，数据源以及任何完成操作所需的结构。每个从句必须起始于一个关键字。最常用的从句及其相关说明如表 9-5 所示。

表 9-5　常用 DCL 语句及其相关说明

子　句	说　明
FROM	指定要对其进行操作的数据源
WHERE	对操作设定一个或多个条件
ORDERBY	对查询结果进行排序
GROUP BY	对查询结果进行分组
HAVING	指定分组的条件

SQL 运算符可分为下面几种类别。

- 比较运算符(大小比较)：>、>=、=、<、<=、<>、!>、!<。
- 范围运算符(表达式值是否在指定的范围)：BETWEEN…AND…、NOT BETWEEN…AND…。
- 列表运算符(判断表达式是否为列表中的指定项)：IN (项 1,项 2……)、NOT IN (项 1,项 2……)。
- 模式匹配符(判断值是否与指定的字符通配格式相符)：LIKE、NOT LIKE。
- 空值判断符(判断表达式是否为空)：IS NULL、NOT IS NULL。
- 逻辑运算符(用于多条件的逻辑连接)：NOT、AND、OR。

SQL 还有一些用来计算的函数，即合计函数。例如，用 AVG 函数计算平均值，用 COUNT 函数返回记录数，用 SUM 函数计算总和，用 MAX 函数计算最大值等。

2. SELECT 语句

SELECT 语句的功能是从现有的数据库中检索数据，将满足一定约束条件的一个或多个表中的字段从数据库中挑选出来，并按一定的分组和排序方法显示出来。

简单的 SQL 查询只包括选择列表、FROM 子句和 WHERE 子句。它们分别说明所查询列，查询的表或视图，以及搜索条件等。例如，下面的语句查询 Customers 表中名为“创新书店”的 CustID 字段和 CustName 字段。

```
SELECT CustID, CustName
FROM Customers
WHERE name='创新书店'
```

选择列表(select_list)指出所查询列，可以是一组列名列表、星号、表达式、变量(包括局部变量和全局变量)等构成。星号代表所有的记录和字段，例如下面语句将显示 Customers 表中所有列的数据：

```
SELECT *
FROM Customers
```

注意：

SELECT 语句中使用 ALL 或 DISTINCT 选项来显示表中符合条件的所有行或删除其中重复的数据行，默认为 ALL。使用 DISTINCT 选项时，对于所有重复的数据行在 SELECT 返回的结果集合中只保留一行。

在选择列表中，可重新指定列标题。如果指定的列标题不是标准的标识符格式时，应使用引号定界符，例如，下面语句使用汉字显示列标题：

```
SELECT  编号= CustID,姓名= CustName
FROM Customers
```

使用 TOP n [PERCENT]选项限制返回的数据行数，TOP n 说明返回 n 行，而 TOP n PERCENT 中，n 表示百分数，指定返回的行数等于总行数的百分之几。例如，下面语句将返回 Customers 表中最前列五分之一的记录：

```
SELECT TOP 20 PERCENT *
FROM Customers
```

3. FROM 子句

FROM 子句指定 SELECT 语句查询及与查询相关的表或视图。在 FROM 子句中最多可指定 256 个表或视图，它们之间用逗号分隔。

在 FROM 子句同时指定多个表或视图时，如果选择列表中存在同名列，这时应使用对象名限定这些列所属的表或视图。例如，在 Customers 和 Orders 表中同时存在 CustName 列，在查询两个表中的 CustName 时应使用下面语句格式加以限定：

```
SELECT CustID,Address, Orders .price
FROM Customers, Orders
WHERE Customers.CustName =Orders. CustName
```

在 FROM 子句中可用以下两种格式为表或视图指定别名：

```
表名 as 别名
表名 别名
```

上面语句可用表的别名格式表示为：

```
SELECT CustID,Address, Orders .price
FROM Customers as 客户资料, Orders 订单资料
WHERE 客户资料.CustName =订单资料. CustName
```

SELECT 不仅能从表或视图中检索数据，还能够从其他查询语句所返回的结果集合中查询数据。

4. WHERE 子句

WHERE 子句用于设置查询条件，过滤掉表中不需要的数据行。可以包含多个表达式，表达式之间要用 AND 和 OR 等运算符连接起来。

例如，下面语句将在订单资料表中查询交易金额为 100~300 的订单明细：

```
SELECT *
FROM Orders
WHERE price BETWEEN 100 AND 300
```

5. ORDER BY 子句

使用 ORDER BY 子句对查询返回的结果进行排序。ORDER BY 子句的语法格式为：

```
ORDER BY {column_name [ASC|DESC]} [,…n]
```

其中 ASC 表示升序，为默认值，DESC 为降序。ORDER BY 不能按 ntext、text 和 image 数据类型进行排序。

ORDER BY 子句中还可包含多个字段，这样记录先按第一个字段排序，然后对值相等的记录再按第二个字段排序。另外，还可以在 ORDER BY 子句中加入表达式，根据计算结果进行排序。例如，下面语句将在订单资料表中先按交易排序：

```
SELECT *
FROM Orders
ORDER BY price desc,CustID ASC
```

6. SELECT…INTO 语句

SELECT…INTO 语句用来从查询结果中建立新表。例如：

```
SELECT CustID, CustName INTO # Customers2
FROM Customers
```

7. DELETE 语句

DELETE 语句的功能是删除 FROM 子句列出的、满足 WHERE 子句条件的一个或多个表中的记录。例如，可以用下面的语句从 Customers 表中删除姓“刘”的记录：

```
DELETE FROM Customers
WHERE CustName LIKE '刘%'
```

9. INSERT INTO 语句

INSERT INTO 语句用于添加一个或多个记录到表中。例如，以下语句向 Customers 表中添加一条新的记录：

```
INSERT INTO Customers
(CustID，CustName，Address，Phone)
VALUES ('KT015'，'陈永超'，'东方红大道 11 号'，'64512458')
```

9. UPDATE 语句

UPDATE 语句用于按某个条件来更新特定表中的字段值。例如，以下语句将 OrderDetails 表中所有单价改为 120：

```
UPDATE OrderDetails SET price=120
```

9.1.4　ODBC 简介

ODBC(Open DataBase Connectivity，开放数据库互连)是 Microsoft 开发的一套读取数

据库的解决方案，它将所有对数据库的底层操作全部隐藏在 ODBC 的驱动程序内核里。对于用户来说，只要构建了一个指向数据库的连接，就可以采用统一的应用程序编程接口(Application Program Interface，API)实现对数据库的读写，或用相同的代码访问不同格式的数据库。

1. ODBC 驱动

ODBC 简化了对数据库的访问，也为程序的跨平台开发和移植提供了极大的方便。ODBC 可对大多数类型数据库提供支持，包括 dBase、Informix、Access、SQL Server 和 Oracle 等，还可对一些其他类的数据库文件提供支持，如文本、Excel 电子表格等。对于一些特殊的数据库，只要安装数据库厂商提供的 ODBC 程序，就能够在程序中直接对数据库进行操作。

ODBC 由应用程序、驱动程序管理器、驱动程序和数据源等部分组成。应用程序通过 ODBC 接口访问不同数据源中的数据，每个不同的数据源类型由一个驱动程序支持。驱动程序管理器为应用程序装入合适的驱动程序，如图 9-15 所示。

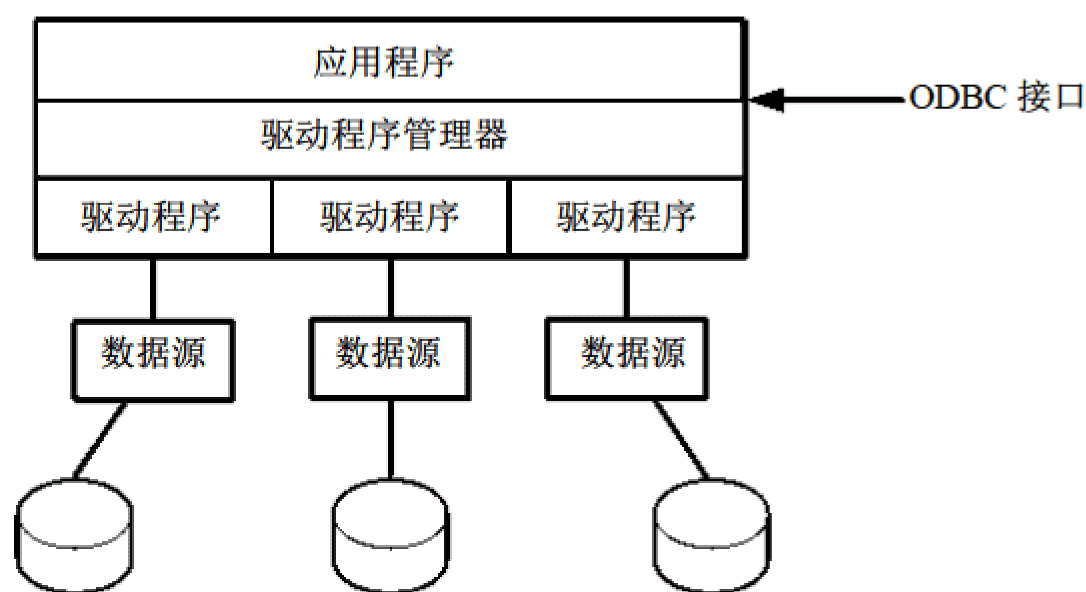


图 9-15　ODBC 的结构

ASP 必须建立与数据库之间的联系才能使用数据库中的数据。要在 ASP 中使用 ADO 对象来操作数据库，首先要创建一个指向该数据库的 ODBC 连接。在 Windows 系统中，ODBC 的连接主要通过 ODBC 数据源管理器来完成。

2. DSN 数据源

数据库驱动程序使用数据源 DSN(Data Source Name)定位和标识特定的 ODBC 兼容数据库，将信息从 Web 应用程序传递给数据库。DSN 包含数据库配置、用户安全性和定位信息，且可以获取 Windows 注册表项或以文本文件格式存储的表格。

构建 ODBC 连接就是创建同数据源的连接，也就是创建 DSN。DSN 就是对数据库的命名连接。一旦创建了指向数据库的 ODBC 连接，同该数据库连接的有关信息就被保存在 DSN 中。在程序中操作数据库也必须通过 DSN 来进行。

DSN 分为用户、系统和文件 3 种类型。用户 DSN 和系统 DSN 将信息存储在 Windows 注册表中，它位于注册表中的如下位置：

```
HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI
```

用户 DSN 只被用户直接使用，只能用于当前计算机中，ASP 不能使用它。系统 DSN 允许所有用户登录到特定服务器上去访问数据库，任何具有权限的用户都可以访问系统 DSN。在 Web 应用程序中访问数据库时，通常都是建立系统 DSN。

文件 DSN 将信息存储在后缀为.dsn 的文本文件中，如果将此文件放在网络的共享目录中，那么可以被网络中的任何一台工作站访问到。

文件 DSN 的优点在于方便移动，因为 DSN 信息保存在独立的文件中，如果希望将整个 Web 应用程序和数据库移动到其他计算机中，只需连同生成的 DSN 文件一起移动即可。而系统 DSN 信息保存在注册表中，所以移动起来就不那么方便。

系统 DSN 的优点在于方便修改，只需简单地修改 Windows 的注册表即可。而对于使用文件 DSN 的用户，则必须每次修改 Global.asa 文件。另外，如果在计算机上许多不同的应用程序须使用同一个 DSN，那么使用系统 DSN 更为方便。

【练习 9-3】以【练习 9-1】创建的数据库为基础，创建一个名为 netdsn 的系统 DSN。

(1) 打开“控制面板”中的“管理工具”窗口，如图 9-16 所示。

(2) 双击“数据源”图标弹出“ODBC 数据源管理器”对话框，选择“系统 DSN”选项卡，如图 9-17 所示。



图 9-16　启动数据源(ODBC)管理器

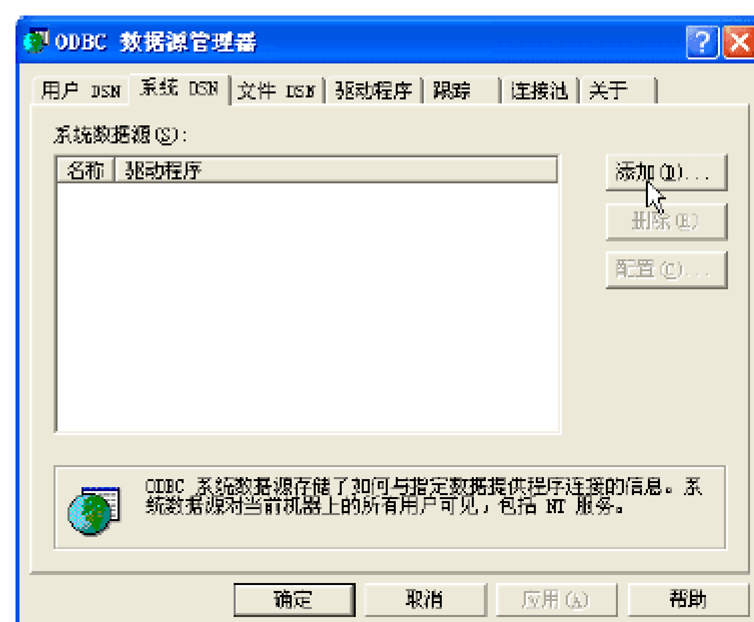


图 9-17　“系统 DSN”选项卡

(3) 单击“添加”按钮，系统弹出“创建新数据源”对话框，选择 Microsoft Access Driver(*.mdb)选项，如图 9-18 所示。

(4) 单击“完成”按钮，在系统打开的“ODBC Microsoft Access 安装”对话框中输入系统 DSN 的名称 netdsn，如图 9-19 所示。

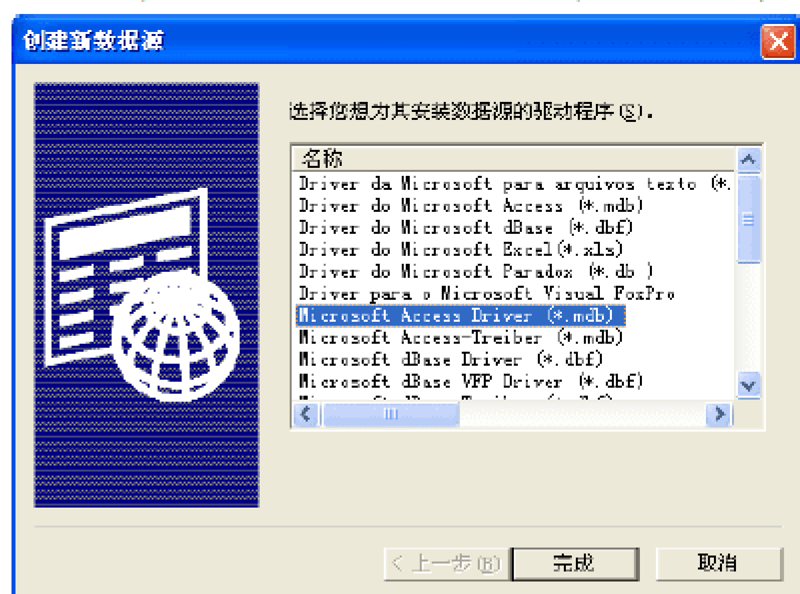


图 9-18　“创建新数据源”对话框

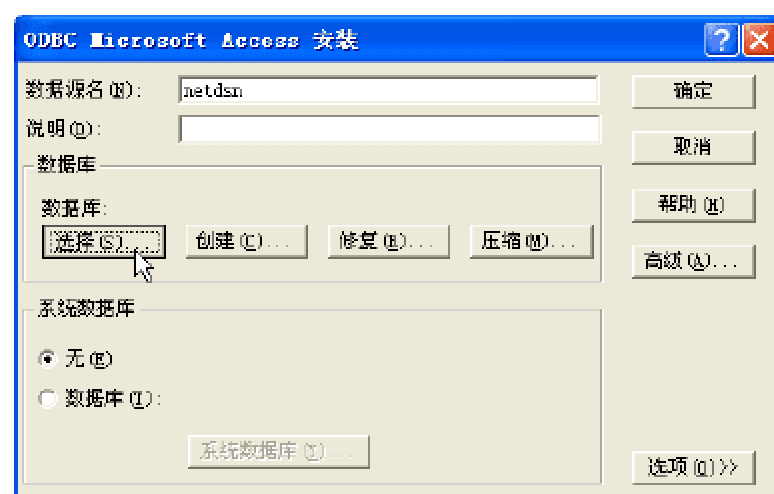


图 9-19　“ODBC Microsoft Access 安装”对话框

(5) 单击“选择”按钮，在系统打开的“选择数据库”对话框中选中【练习 9-1】创建的数据库 db1.msd 后(如图 9-20 所示)，单击“确定”按钮返回“ODBC Microsoft Access 安装”对话框。

(6) 单击“确定”按钮，此时在“ODBC 数据源管理器”对话框中显示刚安装的系统 DSN，如图 9-21 所示。单击“确定”按钮，完成系统 DSN 的安装工作。



图 9-20　选中数据库文件



图 9-21　完成系统 DSN 的安装

9.1.5　OLE DB 简介

OLE DB 提供统一数据访问接口的技术标准。可以访问的数据包括标准关系型数据库中的数据，还包括邮件数据、Web 上的文本或图形、目录服务(Directory Services)，以及主机数据库(如 IMS 和 DB2)、服务器数据库(如 Oracle 和 SQL Server)和桌面数据库(如 Microsoft Access)。

OLE DB 标准的核心内容就是要求以上这些各种各样的数据存储(Data Store)都提供一种相同的访问接口。这种接口封装了各种数据系统的访问操作，使数据的使用者(应用程序)可以使用同样的方法访问各种数据，而不用考虑数据的具体存储地点、格式或类型。OLE DB 还提供了一组标准的服务组件，用于提供查询、缓存、数据更新、事务处理等操作。因此，数据提供方只需实现一些简单的数据操作，使用方就可以获得全部的数据控制能力。

OLE DB 将传统的数据库系统划分为多个逻辑组件，这些组件之间相对独立又相互通信。这种组件模型中的各个部分被冠以不同的名称。

- 数据提供者(Data Provider)：提供数据存储的软件组件，小到普通的文本文件，大到主机上的复杂数据库，或电子邮件存储。
- 数据服务提供者(Data Service Provider)：位于数据提供者之上，从旧的数据库管理系统中分离出来，且独立运行的功能组件，例如查询处理器和游标引擎(Cursor Engine)，这些组件使数据提供者提供的数据以表状数据(Tabular Data)的形式向外表示(不管真实的物理数据是如何组织和存储的)，并实现数据的查询和修改功能。

- 业务组件(Business Component)：利用数据服务提供者，专门完成某种特定业务信息处理，是可以重用的功能组件。分布式数据库应用系统中的中间层(Middle-Tier)就是这种组件的典型例子。
- 数据使用者(Data Consumer)：任何需要访问数据的系统程序或应用程序，除了典型的数据库应用程序之外，还包括需要访问各种数据源的开发工具或语言。

注意：

OLE DB 和 ODBC 标准都是为了提供统一的数据访问接口，ODBC 标准的对象是基于 SQL 的数据源(SQL-Based Data Source)，而 OLE DB 的对象则是范围更为广泛的任何数据存储。所以，符合 ODBC 标准的数据源是符合 OLE DB 标准的数据存储的子集，同时还提供相应的 OLE DB 服务程序(Service Provider)。

OLE DB 标准的具体实现是一组 API 函数，就像 ODBC 标准中的 API 函数一样，不同的是，OLE DB 中的 API 函数是符合 COM 标准、基于对象的。使用 OLE DB 中的 API 函数，可以编写能够访问符合 OLE DB 标准的任何数据源的应用程序，也可以编写针对某种特定数据存储的查询处理程序(Query Processor)和游标引擎(Cursor Engine)，因此 OLE DB 标准实际上是规定了数据使用者和提供者之间的一种应用层的协议(Application-Level Protocol)。

9.1.6　ADO 对象模型

ADO 是应用层的编程接口，通过 OLE DB 提供的接口访问数据，这样各种编程语言都能够编写符合 OLE DB 标准的应用程序。

ADO 封装了 OLE DB 中最常用的一些特性，ADO 可以在 Visual Basic 或 Visual C++ 中使用，也可在服务器端脚本中使用。使用 ADO 时，ASP 应用程序和底层数据库间的关系如图 9-22 所示。

从图 9-22 中可以看出，应用程序既可以通过 ADO 访问数据，也可以直接通过 OLE DB 访问数据，而 ADO 则通过 OLE DB 访问底层数据。

OLE DB 分成两部分，一部分由数据提供者实现，包括一些基本功能，如获取数据、修改数据、添加数据项等；另一部分由系统提供，包括一些高级服务，如游标功能、分布式查询等。这样的层次结构既为数据使用者即应用程序提供了多种选择方案，又为数据提供方简化了服务功能的实现手段，只需按 OLE DB 规范编写一个 COM 组件程序即可，使得第三方发布数据更为简便。而在应用程序方可以得到全面的功能服务，这充分体现了 OLE DB 两层结构的优势。

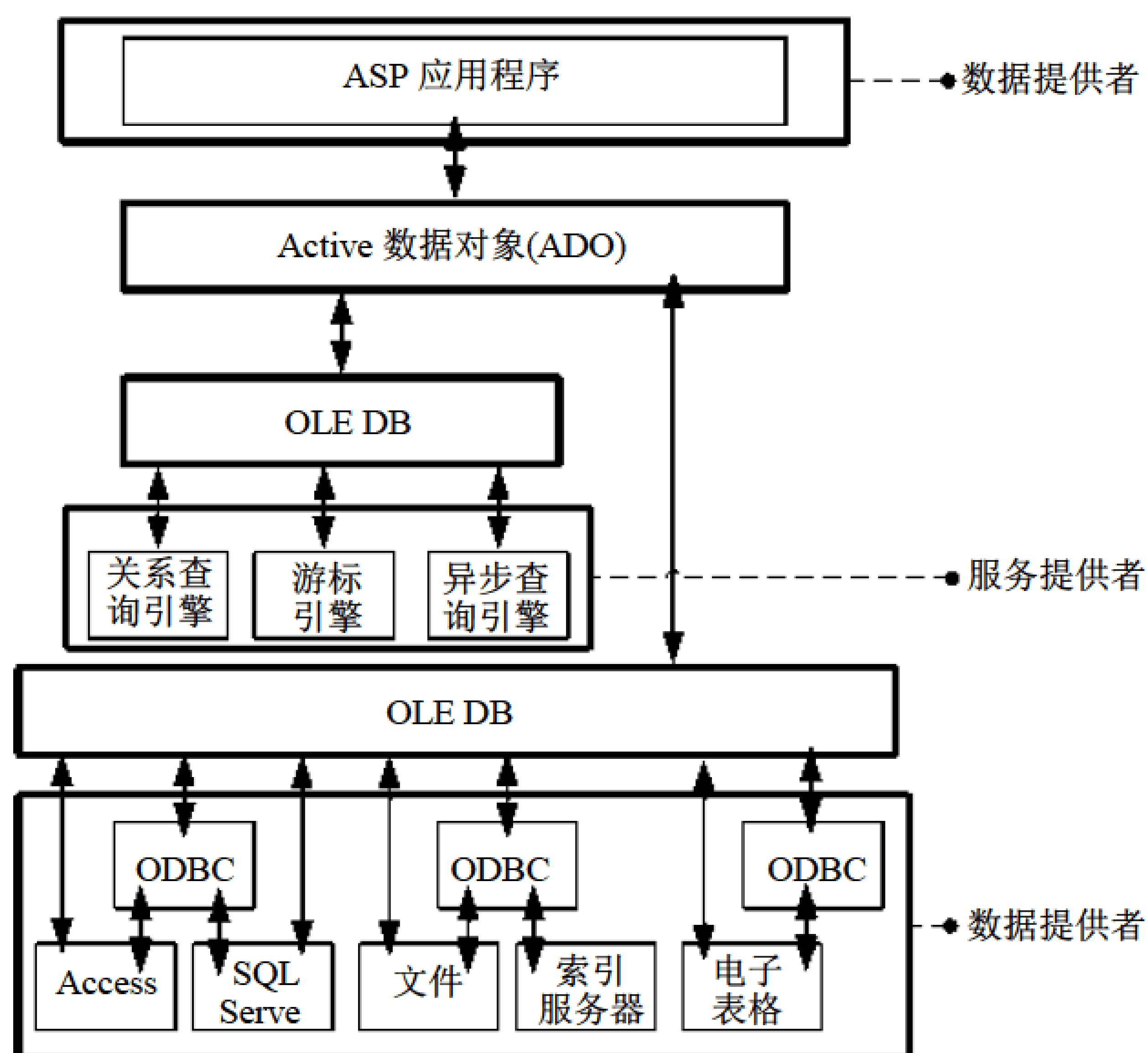


图 9-22　ASP 应用程序和底层数据库间的关系

9.1.7　ADO 对象简介

ADO 实际上是 OLE DB 的应用层接口，这种结构也为一致的数据访问接口提供了很好的扩展性，而不再局限于特定的数据源，因此，ADO 可以处理各种 OLE DB 支持的数据源。ADO 本身由多个对象组成，这些对象分别负责提供各种数据库操作行为，大致上可以分为连接、修改和查询 3 个部分，如图 9-23 所示。

在 ADO 模型中，主体对象只有 Connection、Command 和 Recordset 3 个，一个典型的 ADO 应用使用 Connection 对象建立与数据源的连接，然后用一个 Command 对象给出对数据库操作的命令，例如查询或更新数据等，而 Recordset 用于对结果集数据进行维护或浏览等操作。

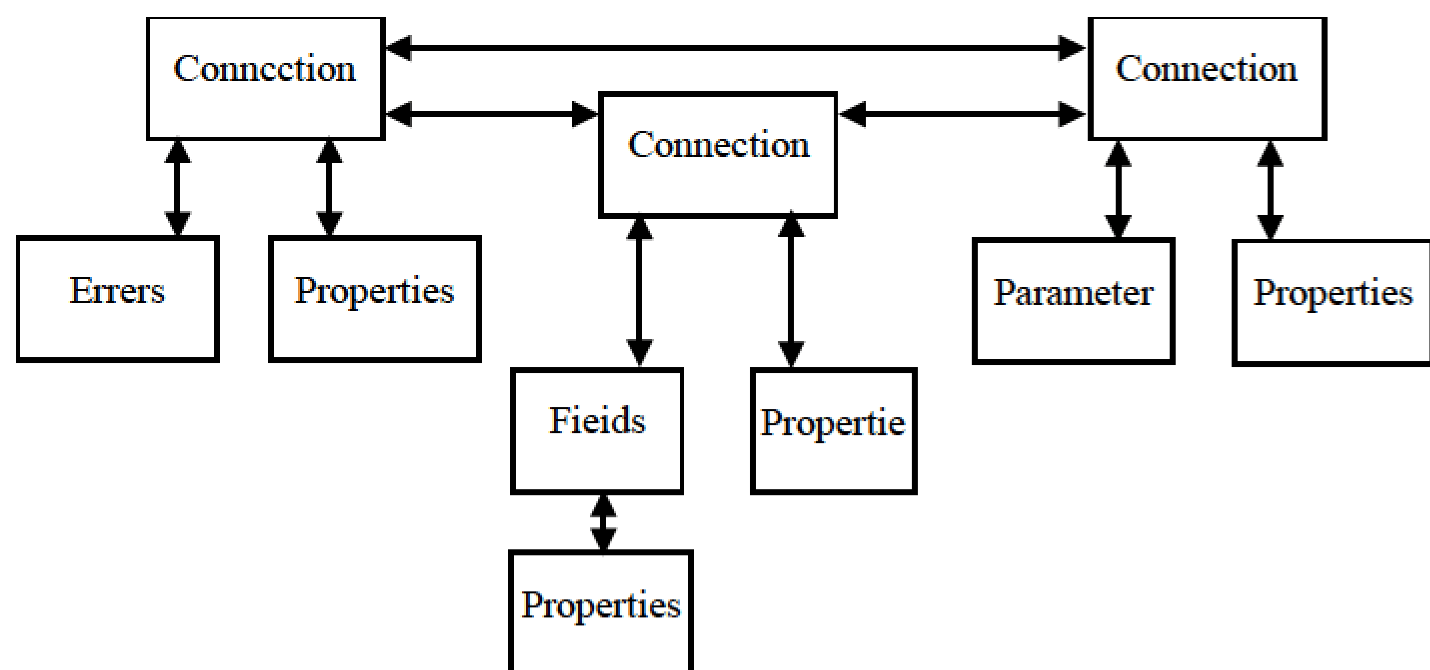


图 9-23　ADO 对象模型

其他 4 个集合对象 Errors、Properties、Parameters 和 Fields 分别对应 Error、Property、Parameter 和 Field 对象，整个 ADO 对象模型由如表 9-6 所示的对象组成。

表 9-6　ADO 中的对象

对　　象	说　　明
Connection 对象	连接。创建与数据库互动所需的连接，任何数据库的操作行为都必须在连接的基础下进行。因此在使用 ADO 之前，首先要创建一个 Connection 对象。必须注意的是，这个动作不是绝对的，ADO 本身会在没有 Connection 对象的情形之下，自行创建所需的连接对象
Command 对象	修改。针对连接的数据库进行数据变动，将用户提供的指令传送到数据库。进行新增、删除或修改资料等变动处理，指令便是用于变动数据的 SQL 语句
Recoreset 对象	查询。从数据库中提取符合特定条件的数据内容。应用程序从 Recoreset 对象取得所要处理的特定数据内容，这些数据可能是某个特定数据表的全部或特定内容，或跨越多个数据所取得的关系型数据，这些数据以二维数组的形式提供
Fields 集合和 Field 对象	Fields 集合处理记录中的各个列。记录集中返回的每一列在 Fields 集合中都有一个相关的 Field 对象。Field 对象使得用户可以访问列名、列数据类型以及当前记录中列的实际值等信息
Parameters 集合和 Parameter 对象	Command 对象包含一个 Parameters 集合。Parameters 集合包含参数化的 Command 对象的所有参数，每个参数信息由 Parameter 对象表示
Properties 集合和 Property 对象	Connection、Command、Recordset 和 Field 对象都含有 Properties 集合。Properties 集合用于保存与这些对象有关的各个 Property 对象。Property 对象表示各个选项设置或其他没有被对象的固有属性处理的 ADO 对象特征
Errors 集合和 Error 对象	Connection 对象包含一个 Errors 集合。Errors 集合包含的 Errors 对象给出了关于数据提供者出错时的扩展信息

9.2　Connection 对 象

Connection 对象又称连接对象，用来和数据库建立连接。Connection 对象建立连接后，才可以利用 Command 对象或 Recordset 对象对数据库进行各种操作。

9.2.1　与数据库建立连接

建立 Connection 对象是采用 Server 对象的 CreateObject 方法进行的，其语法结构如下：

```
Set Connection 对象= Server.CreateObject("ADODB.Connection")
```

其中，ADODB.Connection 为所要创建的 ADO 连接对象。另外，还可用<OBJECT>标记来创建 Connection 对象，例如：

```
<OBJECT RUNAT=Server ID=cn PROGID="ADODB.Connection"></OBJECT>
```

注意：

在没有明确建立 Connection 对象的情形之下，ADO 本身也会自行创建所需的连接对象，但这样将无法利用 Connection 对象的许多功能。

1. 用 DSN 连接数据库

一旦连接对象创建后，便可以通过这个对象来连接数据库，这时只需直接引用 Connection 对象的 Open 方法即可。其语法结构如下：

```
Connection 对象.Open 连接字符串或变量
```

在“连接字符串或变量”中各参数的意义如表 9-7 所示。

表 9-7　Connection 对象的 Open 方法的参数

参　数	说　明
Dsn	ODBC 数据源名称
User	数据库登录账号
Password	数据库登录密码
Driver	数据库的类型(驱动程序)
Dbq	数据库的物理路径
Provider	数据提供者
Data Source	数据源

如果用到两个以上的参数，中间用分号隔开，顺序没有关系。有些参数不能同时使用，例如用了 Driver，一般就不用 Provider；用了 Dsn，也就不用 Driver 和 Provider，如下例所示：

```
<%cn.Open "booknetdsn","sa",""%>
```

也可以先设置 ConnectionString 属性，再调用 Open 方法。例如：

```
<%
cn.ConnectionString= "DSN=booknetdsn; "
cn.Open
%>
```

或直接将连接字符串作为参数来调用：

```
<%cn.Open "DSN=booknetdsn;UID=sa;PWD="%>
```

注意：

创建有 ODBC 数据源的连接方法书写简便，也不容易出错。更重要的是不管数据库放在哪里，只要对数据源重新设置即可，不需更改程序代码。如果要将程序移植到另外的服务器上，则需要重新设置数据源。

2. 创建基于 OLE DB 连接字符串的连接

如果在"ODBC 数据源管理器"中没有建立 DSN，那么 ADO 便使用提供的 OLE DB 连接字符串来识别 OLE DB 提供者，并将提供者指向数据源，如下例所示：

```
<%
cn.Open "Provider=SQLOLEDB.1;User ID=sa;Password=;"&_
"Initial Catalog=booknet;Data Source=zsh"
%>
```

其中，Provider 指定数据提供者，Initial Catalog 指向在 SQL Server 上的待访问的数据库，Data Source 指定 SQL Server 的计算机名或 IP 地址。

对于 Access 数据库也可以采用这种方式进行连接，如下例所示：

```
<%cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=e:\book.mdb"%>
```

3. 用 ODBC 连接字符串连接数据库

SQL 数据库也可以不设立数据源，要用到 Driver 和 Dbq 两个参数，如下例所示：

```
<%cn.Open "Driver=;Database=booknet;Server=(local);UID=sa;PWD="%>
```

其中，Driver 指定数据源驱动程序的名称，SQL Server 使用{SQL Server}来进行标识；Database 指定所请求的默认数据库；Server 指定数据源服务器的名称，设为(local)时，表示使用 SQL Server 的本地副本。

注意：

创建没有 ODBC 数据源连接的应用程序在移植到别的服务器上后，就可以立即使用，因此它是应用程序中比较常用的与数据库连接的方法。

对于其他类型的数据库，它们的连接方法如表 9-8 所示。

表 9-8　ODBC 连接字符串

数据源驱动器	ODBC 连接字符串
Microsoft Access	Driver={Microsoft Access Driver(*.mdb)};DBQ=指向.mdb 文件的物理路径
Oracle	Driver={Microsoft ODBC for Oracle};SERVER=指向服务器的路径
Microsoft Excel	Driver={Microsoft Excel Driver(*.xls)};DBQ=指向.xal 文件的物理路径;DriverID=278
Microsoft Excel 97	Driver={Microsoft Excel Driver(*.xls)};DBQ=指向.xal 文件的物理路径;DriverID=790

(续表)

数据源驱动器	ODBC 连接字符串
Paradox	Driver={Microsoft Paradox Driver(*.db)};DBQ=指向.db 文件的物理路径;DriverID=26
文本	CDriver={Microsoft Text Driver(*.txt;*.csv)};DefaultDir=指向.txt 文件的物理路径
Visual FoxPro(带有一个数据库容器)	Driver={Microsoft Visual FoxPro Driver};SourceType=DBC;SourceDb=指向.dbc 文件的物理路径
Visual FoxPro(不带数据库容器)	Driver={Microsoft Visual FoxPro Driver};SourceType=DBF;SourceDb=指向.dbf 文件的物理路径

【练习 9-4】以本章【练习 9-1】创建的数据库 db1.mdb 和【练习 9-3】创建的系统 DSN 为基础，创建 Connection 对象并与数据库建立连接，读取数据库指定表中的第一条记录，程序运行效果如图 9-24 所示。

(1) 完成【练习 9-3】的操作后，双击打开 C:\Inetpub\wwwroot\AcDatabase 文件夹中保存的数据库 db1.mdb(如图 9-25 所示)，然后在该数据库的数据表“表 1”中输入如图 9-26 所示的数据。



图 9-24　程序运行效果

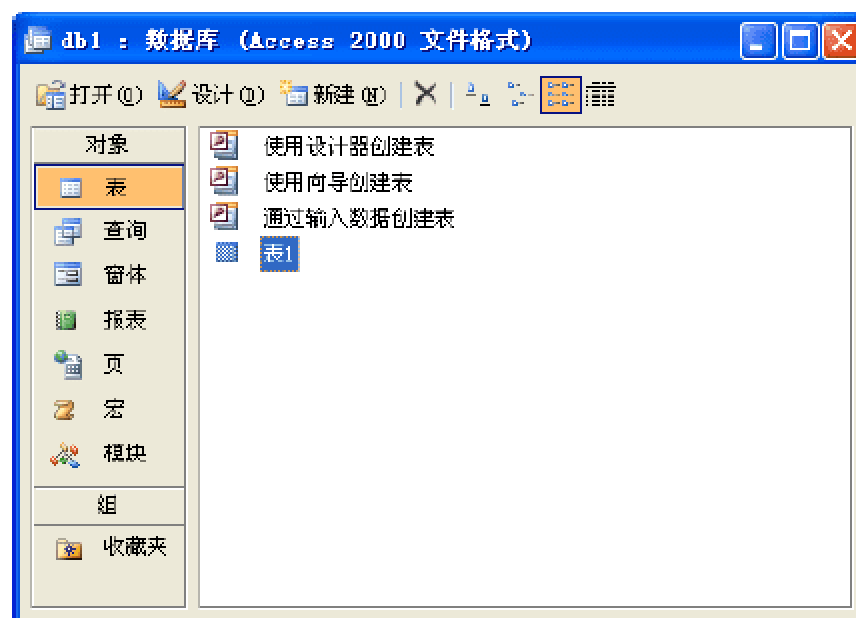


图 9-25　打开 db1.mdb 数据库

(2) 完成数据的输入后，选择“文件”|“保存”命令，将数据表“表 1”保存，然后关闭 Access 数据库。

(3) 选择“开始”|“所有程序”|“附件”|“记事本”命令，打开记事本工具，输入以下代码(如图 9-27 所示):

```
<Center><H4>数据表"表 1"中的第一条记录</H4></Center>
<Hr>
<%
Set conn= Server.CreateObject("ADODB.Connection")     '创建 Connection 对象 conn
conn.Open "netdsn","sa",""                             '使用 DSN 建立 conn 与数据库的连接
Set rs = Server.CreateObject ("ADODB.Recordset")       '创建 Recordset 对象 rs
```

```
sql = "Select * from  表 1"                       '指定一个 SQL 查询语句
rs.Open sql,conn,1,1                              'Recordset 对象使用 SQL 查询语句打开数据库
Response.Write   rs.Fields("C-Id")&"<Br>"         '显示数据库中的内容
Response.Write   rs.Fields("C-Name")&"<Br>"
Response.Write   rs.Fields("C-address")&"<Br>"
Response.Write   rs.Fields("C-Tel")&"<Br>"
Response.Write   rs.Fields("C-Mail") %>
```

(4) 选择"文件"|"另存为"命令，将文件以名 Connection.asp 保存至 C:\Inetpub\wwwroot 文件夹中。

(5) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Connection.asp 并按 Enter 键，效果如图 9-24 所示。

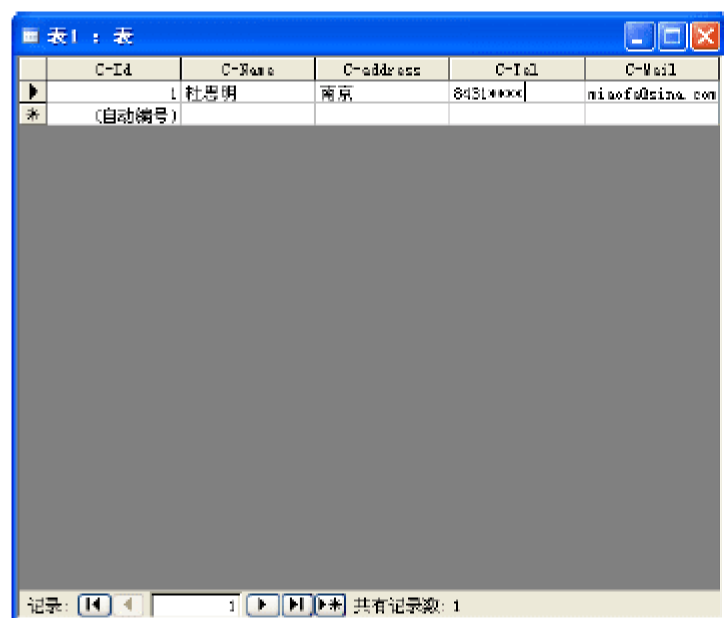


图 9-26　输入数据

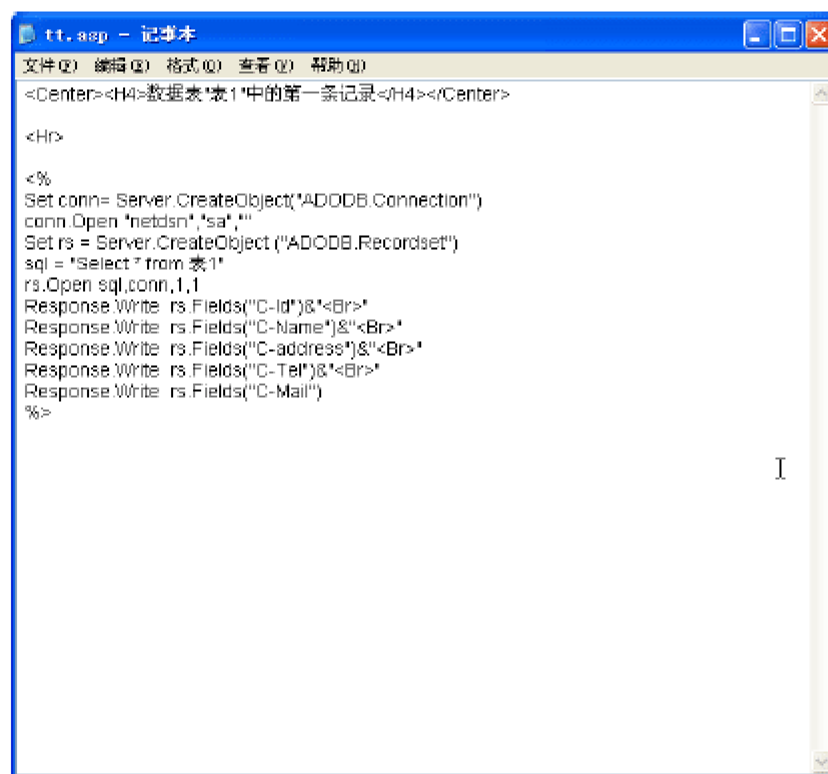


图 9-27　Connection.asp 的代码

【练习 9-5】以【练习 9-4】在数据表"表 1"中输入的数据为基础，创建一个具有多种与数据库连接方式的连接文件，并在 ASP 程序包含它显示数据库中表的内容，程序运行效果如图 9-28 所示。

(1) 创建一个名为 Conn.asp 的文件，其代码如下所示：

```
<%
dim Conn
Set conn= Server.CreateObject("ADODB.Connection")
'使用 DSN 与数据库建立连接的第 1 种方法
conn.Open "netdsn","sa",""
'使用 DSN 与数据库建立连接的第 2 种方法
conn.ConnectionString = "DSN=netdsn;UID=sa;PWD=;"
conn.Open
'使用 DSN 与数据库建立连接的第 3 种方法
conn.Open "dsn=netdsn;UID=sa;PWD=;"
'使用 OLE DB 字符串与数据库建立连接
conn.Open "Provider=SQLOLEDB.1;User ID=sa;Password=;Initial Catalog=booknet;Data
Source=local"
```

```
'使用 ODBC 字符串与数据库建立连接
dim strConn
strConn = "Driver={SQL
    Server};Database=booknet;Server=ZHANGSHIHUA\TEST;UID=sa;PWD=;"
conn.Open strConn
%>
```

(2) 将 Conn.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 创建一个名为 Connection2.asp 的文件，其代码如下所示(如图 9-29 所示):

```
<Title>数据库连接</Title>
<!--#include file="conn.asp" -->
<Center><H4>显示数据库中表的内容</H4></Center>
<Hr>
<%
Set rs = Server.CreateObject ("ADODB.Recordset")
sql = "Select * from 表 1"
rs.Open sql,conn,1,1
With Response
  if rs.EOF then
    Response.Write "现在数据库为空！"
  else
    Response.Write "<TABLE BORDER=1 CELLSPACE=0 CELLPADDING=5>" &_
     "<TR HEIGHT=12><TD WIDTH=50><B> 编号 </B></TD>" &_
     "<TD WIDTH=100><B> 名称 </B></TD>" &_
     "<TD WIDTH=200><B> 地址 </B></TD>"&_
     "<TD WIDTH=80><B> 电话 </B></TD></TR>"
  end if
  do Until rs.EOF
    Response.Write "<TR HEIGHT=12><TD WIDTH=50>" & rs.Fields("C-Id")& "</TD>" &_
    "<TD WIDTH=100>" & rs.Fields("C-Name") & "</TD>" &_
    "<TD WIDTH=200>" & rs.Fields("C-address") & "</TD>"&_
    "<TD WIDTH=80>" & rs.Fields("C-Tel") & "</TD></TR>"
    rs.MoveNext
 loop
 Response.Write "</TABLE>"
End With
rs.close
Set rs = Nothing
conn.close
set conn=Nothing
%>
```

(4) 将 Connection2.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(5) 启动 IE 浏览器后，在地址栏中输入 http://localhost/Connection2.asp 并按 Enter 键，效果如图 9-28 所示。

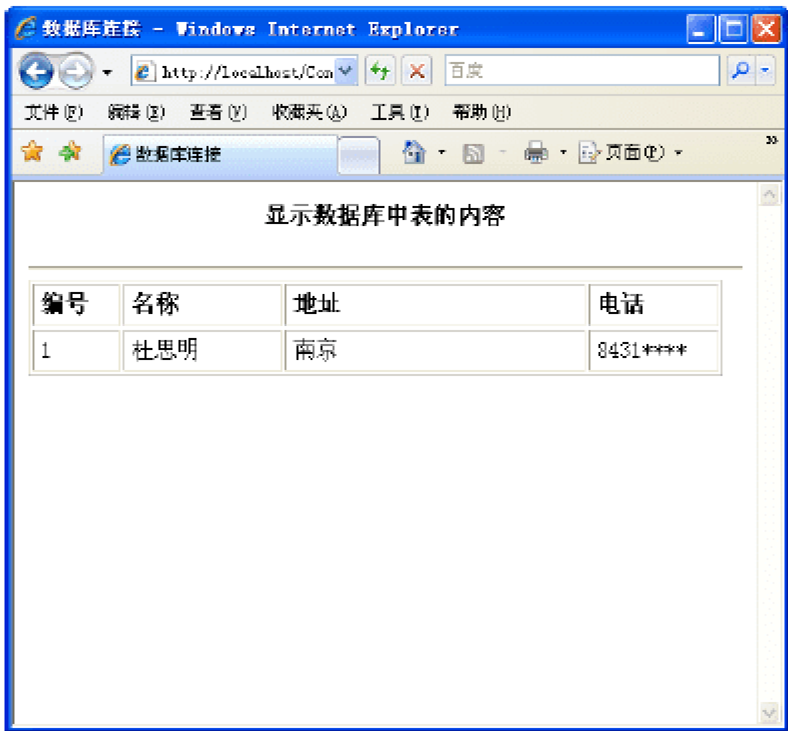


图 9-28　Connection2.asp 的效果

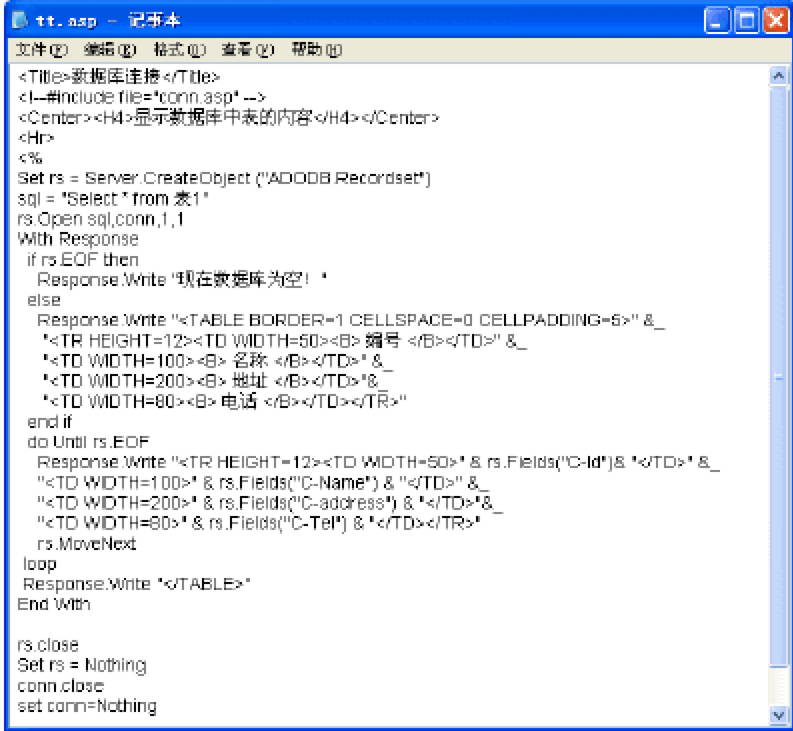


图 9-29　Connection2.asp 的代码

9.2.2　Connection 对象的属性

Connection 对象的属性及其相关说明如表 9-9 所示。

表 9-9　Connection 对象的属性

属　　性	说　　明
Attributes	设置 Connection 对象控制事务处理时的行为
CommandTimeout	Execute 方法的最长执行截止时间
ConnectionString	指定 Connection 对象的数据库连接信息
ConnectionTimeout	Open 方法与数据库连接的执行截止时间
CursorLocation	控制光标的类型。确定是使用客户端(adUseClient)游标引擎，还是使用服务器端(adUseServer)游标引擎。默认值是 adUseServer
DefaultDatabase	在数据提供者提供多个数据库的情况下，如果 ConnectString 中未指定数据库名称，就使用这里所指定的名称
IsolationLevel	指定和其他并发事务交互时的行为或事务
Mode	指定对 Connection 的读写权限
Provider	如果 ConnectionString 中未指定 OLE DB 数据或服务提供者的名称，就使用这时指定的名称。默认值是 MSDASQL(Microsoft OLE DB Provider for ODBC)
State	指定连接的状态。若是 0 或 adStateClosed，则连接是关闭的；若是 1 或 adStateOpen，则连接是打开的
Version	返回 ADO 版本号

下面对表 9-9 中常用的几个 Connection 对象的属性进行介绍。

1. Attributes 属性

Attributes 属性通过两个常数确定当前事务失败或成功后是否自动开始一次新的事务。如果设为 131072 或 ADO 常量 adXactCommitRetaining，则调用 CommitTrans 方法时自动启动一次新的事务；如果设为 262144 或 ADO 常量 adXactAbortRetaining，则调用 RollbackTrans 方法后，自动开始一次新的事务。如果要达到两种效果，则该属性应设为两者的和。

2. CommandTimeout 属性

CommandTimeout 属性用来设置 Connection 对象的 Execute 方法的最长执行时间。其默认值为 30s。如果超过时间未完成命令，则终止命令并产生一个错误。命令无法在指定时间内执行完成，可能是因为网络延时或服务器负载过重而无法及时响应造成的。如果将该属性设为 0，则无限期地等待直到执行完成。如下例将把 CommandTimeout 的最长时间设置为 60s：

```
<%db. CommandTimeout=60%>
```

3. ConnectionString 属性

ConnectionString 属性指定数据提供者或服务提供者打开数据源连接所需要的特定信息。除了可以使用 Connection 对象的 Open 方法来打开数据库外，还可以使用 Connection 对象的 ConnectionString 属性来打开数据库，如下例所示：

```
<%
Dim db
Set db=Server.CreatObject("ADODB.Connection")
db.ConnectionString="Dbq="&Server.Mappath("db1.mdb")&";_
Driver={Microsoft Access Driver(*.mdb)}"
db.Open
%>
```

4. ConnectionTimeout 属性

确定 ADO 试图与一个数据源建立连接时的最大连接时间，默认值是 15s。如果超过时间未完成连接，则终止连接并产生一个错误。如果将该属性设为 0，则一直等待直到连接成功为止。如下例将把 ConnectionTimeout 的默认值设置为 30s：

```
<%db. ConnectionTimeout=30%>
```

5. Mode 属性

Mode 属性用来设置连接数据库的权限，利用该属性就可以在打开数据库时限制数据

库的连接方式，如只读或只写。如果不进行设置，可具有对数据库进行读写操作的权限，Mode 属性的取值及其相关说明如表 9-10 所示。

表 9-10　Mode 属性的取值及其相关说明

MODE 属性	整　数　值	说　　明
adModeUnknown	0	未指定数据源的连接许可权(默认值)
adModeRead	1	连接是只读的
adModeWrite	2	连接是只写的
adModeReadWrite	3	连接是可读写的
adModeShareDenyRead	4	拒绝其他用户打开数据源的读连接
adModeShareDenyWrite	8	拒绝其他用户打开数据源的写连接
adModeShareExclusive	12	以独占方式打开数据源
adModeShareDenyNone	16	其他用户不能以任何方式打开连接

9.2.3　Connection 对象的方法

Connection 对象的方法及其相关说明如表 9-11 所示。

表 9-11　Connection 对象的方法

方　　法	说　　明
Open	建立 Connection 对象和数据库之间的连接
Close	关闭 Connection 对象和数据库之间的连接
Execute	执行数据库查询(可以执行各种操作)
Cancel	取消未执行完的异步 Execute 或 Open 方法
BeginTrans	开始事务处理
CommitTrans	提交一个事务处理结果
RollbackTrans	取消一个事务处理结果

下面对表 9-11 中常用的几个 Connection 对象的方法进行介绍。

1. Open 方法

Open 方法用来建立 Connection 对象和数据库之间的连接。只有用 Open 方法和数据库建立连接后，才可以继续进行各种操作。

2. Close 方法

Close 方法用来关闭一个已打开的 Connection 对象及其相关的各种对象。它的主要作用是切断 Connection 对象与数据库之间的连接通道。当该通道被关闭后，所有依赖该 Connection

对象的 Command 或 Recordset 对象也将立即被切断，方法如下：

```
<%
db.Close
Set db=nothing
%>
```

注意：

db.Close 用来关闭连接，这样可以释放与连接有关的系统资源。Connection 对象本身并没有释放，还可以更改该对象的属性并重新打开。如果要从内存中完全释放 Connection 对象占用的资源，可以将其设为 Nothing。

3. Execute 方法

Execute 方法执行指定的查询、SQL 语句、存储过程或数据提供者指定的文本，其语法结构有两种：

```
Set Recordset 对象=Connection 对象. Execute(SQL 字符串)
```

或

```
Connection 对象.Execute(SQL 字符串)
```

注意：

当对数据库查询显示记录时常采用第一种 Execute 执行方法，它将返回一个 Recordset 对象；而执行添加、删除和更新操作时常采用第二种 Execute 执行方法，它不返回 Recordset 对象。

在不返回 Recordset 对象时，可使用一个 number 参数来返回此操作影响的记录条数，如下例所示：

```
<%
strSql="Delete From users Where name='陈功'"
db.Execute strSql,number
Response.Write "共删除"&number&"条记录"
%>
```

4. BeginTrans 方法

BeginTrans 方法用于开始一个事务处理。其语法结构如下：

```
Connection 对象. BeginTrans
```

所有的数据库操作都属于事务处理。当开始一个事务处理后，就打开 Web 页面与数据库的事务处理通道，此时可以从 Web 页面上直接更新数据库内容。但只有在提交事务处理结果后，数据库的内容才能被真正更新，否则，所有的操作都无效。

9.3　Command 对象

Command 对象定义将对数据源执行的指定命令，这些命令可以是 SQL 语句、表名、存储过程或其他数据提供者支持的文本格式。Command 对象的作用相当于一个查询，使用它可以查询数据库并返回记录集，也可执行大量操作或处理数据库结构。

用 Command 对象执行查询的方式与用 Connection、Recordset 对象执行查询的方式一样，但使用 Command 对象可以改善查询。用 Command 对象的参数查询，可先在数据源上准备一种查询方式，然后用不同的值来重复执行查询，以避免重复发出类似的 SQL 查询语句。

9.3.1　创建 Command 对象

创建 Command 对象的语法结构如下：

```
Set Command 对象=Server.CreatObject("ADODB.Command")
```

然后，可用 ActiveConnection 属性指定要利用的 Connection 对象名称，语法如下：

```
Command 对象.ActiveConnection=Connection 对象
```

1. 通过 Connection 对象创建 Command 对象

每个 Command 对象都有一个相关联的 Connection 对象。在创建 Command 对象之前，一般应该先建立 Connection 对象。

通过 Connection 对象创建 Command 对象，如下例所示。

```
<%
Dim conn,cmd
'创建 Connection 对象 conn
Set conn= Server.CreateObject("ADODB.Connection")
'使用 DSN 建立 conn 与数据库的连接
conn.Open "booknetdsn","sa",""
'创建 Command 对象 cmd
Set cmd= Server.CreateObject("ADODB.Command")
'将 Connection 对象 conn 指定给 Command 对象 cmd
cmd.ActiveConnection=conn
%>
```

2. 直接创建 Command 对象

对于 Command 对象，也可以不先创建 Connection 对象就直接使用，只需设置 Command 对象的 ActiveConnection 属性为一个连接字符串即可。此时，ADO 会自行创建一个隐含的

Connection 对象，但并不给它分配对象变量。

注意：

创建 Command 对象的过程中，如果没有把 ActiveConnection 属性设置为一个明确的 Connection 对象，即使使用相同的连接字符串，ADO 也会为每个 Command 对象创建一个新的连接。

不通过 Connection 对象直接创建 Command 对象，如下例所示。

```
<%
Dim cmd
Set cmd= Server.CreateObject("ADODB.Command ")
cmd. ActiveConnection= "addr"
%>
```

9.3.2　Command 对象的属性

Command 对象的属性及其相关说明如表 9-12 所示。

表 9-12　Command 对象的属性

属　　性	说　　明
ActiveConnection	指定 Connection 的连接对象
CommandText	指定数据库的查询信息
CommandType	指定数据查询信息的类型
CommandTimeout	指定 Command 对象的 Execute 方法的最长执行时间
Prepared	指定数据查询信息是否要先编译和存储

下面对表 9-12 中常用的几个 Command 对象的属性进行介绍。

1. ActiveConnection 属性

ActiveConnection 属性设置或返回 Command 对象的连接信息，该属性可以是一个 Connection 对象或连接字符串。其语法结构为：

```
Command 对象.ActiveConnection=Connection 对象
```

如果没有明确建立 Connection 对象，则其语法结构为：

```
Command 对象.ActiveConnection=数据源名称字符串
```

2. CommandText 属性

CommandText 属性设置或返回数据源的命令串，该命令串可以是 SQL 语句、表、存

储过程或数据提供者支持的任何特殊有效的命令文本。其语法结构如下：

```
Command 对象. CommandText=SQL 语句或数据表名或查询名或存储过程名
```

注意：

如果 CommandText 属性指定的是数据表名，则将查询和返回整个数据表中的所有内容。

3. CommandType 属性

CommandType 属性用于指定 Command 对象中数据查询信息的类型，其语法结构如下：

```
Command 对象.CommandType=类型值
```

CommandType 属性的取值及其相关说明如表 9-13 所示。

表 9-13　CommandType 属性的取值及其相关说明

CommandType 属性	整　数　值	说　　明
AdCmdText	1	SQL 命令类型
AdCmdTable	2	数据表名
AdCmdStoredProc	4	查询名或存储过程名
AdCmdUnknown	8	未知的。CommandText 参数类型无法确定
AdExecuteNoRecords	128	不返回记录集的命令或存储过程
AdCmdFile	256	已存在的记录集的文件名
AdCmdTableDirect	512	CommandText 是一个表，在查询中返回该表的全部行和列

为 Command 对象指定 CommandType 值，如下例所示：

```
<%
……
Set cmd= Server.CreateObject("ADODB.Command ")
cmd. ActiveConnection=conn
cmd.CommandType=1
cmd.CommandText="Select * From users"
cmd.CommandType=2
cmd.CommandText="users"
%>
```

注意：

在未指定 CommandType 值的情况下，系统会自行进行判定查询信息的类型。指定 CommandType 值可以节省系统判定过程的时间，加快系统运行的速度。

4. CommandTimeout 属性

CommandTimeout 属性设置执行一个 Command 对象时的等待时间，默认值是 30s。如果在这个时间内 Command 对象没有执行完，则终止命令并产生一个错误。

5. Prepared 属性

Prepared 属性指出在调用 Command 对象的 Execute 方法时，是否将查询的编译结果存储下来。如果将该属性设为 True，则会编译并保存查询结果，这样将影响第一次的查询速度，但一旦数据提供者编译了 Command 对象，数据提供者在以后的查询中将使用编译后的版本，从而极大地提高速度。其语法结构如下：

```
Command 对象. Prepared=布尔值
```

9.3.3　Command 对象的方法

Command 对象的方法及其相关说明如表 9-14 所示。

表 9-14　Command 对象的方法

方　法	说　明
Execute	执行数据库查询(可以执行各种操作)
CreateParameter	用来创建一个 Parameter 子对象
Cancel	取消一个未确定的异步执行的 Execute 方法

9.3.4　使用 Command 对象的方法

使用 Command 对象有几个重要的步骤，创建 Command 对象，指定对象数据库连接，指定 SQL 指令和引用 Execute 方法。

创建 Command 对象和连接数据库一样，运用 Command 对象之前首先必须引用 CreateObject 对象，创建其对象实体，设定对象识别名称，如下：

```
Dim objCommand
Set objCommand =Server.CreateObject("ADODB.Command")
```

其中，objCommand 为所要创建的指令对象名称，应用程序在创建对象实体后，以此作为名称识别。

- 指定对象数据库连接：Command 对象以特定连接为基础，针对连接的数据库进行存取操作，应用程序必须设置 Command 对象所要存取的数据库连接对象。
- SQL 指令：Command 对象本身并没有具备数据变动的功能，其主要功能在于将指定的 SQL 语句传送至数据库，由数据库根据传送过来的 SQL 进行数据存取操作，

ASP 网页则负责将特定的 SQL 指定给 Command 对象。

- 引用 Excute 方法：当 Command 对象设定完成后，最后只需引用方法 Execute，即可将指定的 SQL 通过连接对象传送至伺服端数据库作处理。

9.3.5　参数查询

如果要创建一个使用多次但每次使用不同值的查询，那么应在查询中使用参数，即创建参数查询。参数是查询时所提供值的占位符，它将 WHERE 子句中固定值用“?”来代替，称作占位符号。这样就避免了在每次查询中重新建立 SQL 查询语句。

一个 Parameter 对象就是一个参数，Parameters 集合就是若干个参数的集合。Parameter 对象和 Parameters 集合都有各自的属性和方法。

1. Parameters 集合的属性和方法

Command 对象包含一个 Parameters 集合。Parameters 集合包含参数化的 Command 对象的所有参数，每个参数信息由 Parameter 对象表示。Parameters 集合的属性和方法及其相关说明如表 9-15 所示。

表 9-15　Parameters 集合的属性和方法

名　称	说　明
Count 属性	返回 Command 对象的参数个数
Append 方法	增加一个 Parameter 对象到 Parameters 集合中
Delete 方法	从 Parameters 集合中删除一个 Parameter 对象
Item 方法	取得集合内的某个对象
Refresh 方法	重新整理 Parameters 数据集合

2. 创建 Parameter 对象

要执行一个参数查询，必须先调用 CreateParameter 方法创建一个 Parameter 对象，然后调用 Append 方法将其添加到 Parameters 集合中，再将值赋给参数。创建 Parameter 对象的语法结构如下：

```
Set Parameter 对象=Command 对象.CreateParameter(Name,Type,Direction,Size,Value)
```

其中，Name 表示参数名，Type 表示参数类型，Direction 表示参数的数据流向，Size 表示字符参数串长度，Value 表示参数的值。

创建 Parameter 对象的过程中，参数的类型由 Type 来定义，它的取值及其相关说明如表 9-16 所示。

表 9-16　Type 的取值范围

常　　量	值	说　　明
AdBigInt	20	8 位符号整数
AdBinary	128	二进制值
AdBoolean	11	布尔值
AdBSTR	8	以空值结束的 Unicode 字符串
AdChar	129	字符串值
AdCurrency	6	货币值，8B 长
AdDate	7	日期值
AdDBDate	133	日期值，格式是：yyyymmdd
AdBDTime	134	时间值，格式是：hhmmss
AdDBTimeStamp	135	日期时间值，格式是：yyyymmddhhmmss
AdDecimal	14	有固定精度的数值
AdDouble	5	双精度浮点值
AdEmpty	0	无指定值
AdError	10	32 位错误码
AdGUID	72	全局唯一指示符
AdIDispatch	9	指向 OLE 对象的接口指针
AdInteger	3	4 位符号整数
AdIUnknown	13	指向 OLE 对象的 IUnknown 接口指针
AdLongarBinary	205	长整型二进制(仅用于 Parameter 对象)
AdLongarChar	201	长字符串值(仅用于 Parameter 对象)
AdLongarWChar	203	以空值结束的长字符串值(仅用于 Parameter 对象)
AdNumeric	131	有固定精度的数值
AdSingle	4	单精度浮点值
AdSmallInt	2	两位符号整数
AdTinyInt	16	一位符号整数
AdUnsignedBigInt	21	8 位无符号整数
AdUnsignedInt	19	4 位无符号整数
AdUnsignedSmallInt	18	两位无符号整数
AdUnsignedTinyInt	17	一位无符号整数
AdUserDefined	132	用户定义的变量
AdvarBinary	204	二进制值(仅用于 Parameter 对象)
AdvarChar	200	字符串值(仅用于 Parameter 对象)
AdVariant	12	OLE 的 Variant 类型
AdVarWChar	202	以空值结束的 Unicode 字符串(仅用于 Parameter 对象)
AdWChar	130	以空值结束的 Unicode 字符串

创建 Parameter 对象的过程中，参数的数据流向由 Direction 来定义，它的取值及其相关说明如表 9-17 所示。

表 9-17　Direction 的取值范围

常　　量	值	说　　明
AdParamInput	1	输入参数，即传送数据给一个存储过程
AdParamOuput	2	输出参数，即得到 Command 对象执行后的输入值
AdParamInputOutput	3	输入和输出参数，即传送并接收数据
AdParamReturnvalue	4	返回值，用来读取从存储过程返回的状态值

3. Parameter 对象的属性

Parameter 对象表示基于带参数的查询或存储进程的 Command 对象相关的参数。Parameter 对象的一些属性是从传递给 Command 对象 CreateParameter 方法的参数那里继承而来的，Parameter 对象的属性及其相关说明如表 9-18 所示。

表 9-18　Command 对象的 CreateParameter 方法的参数意义

参　　数	说　　明
Name	参数名称
Type	参数类型
Direction	参数方向，传入还是传出
Size	参数大小，指定最长字节，可以省略
Value	参数值
Attributes	指定该参数的数值性质

其中 Attributes 参数用来指定参数值的性质，其取值及其相关说明如表 9-19 所示。

表 9-19　Attributes 参数的取值范围

常　　量	值	说　　明
AdParamLong	128	允许有相当大的数值
AdParamNullable	64	允许 NULL 值
AdParamSigned	16	允许数值有正负符号

4. Parameter 对象的方法

Parameter 对象只有一个 AppendChunk 方法，用来处理传递给一个参数的长文本或二进制数据。它允许把一个长文本或二进制信息加入到 Parameter 对象的末尾，其语法结构如下：

Parameter 对象.AppendChunk(长文本或二进制数据)

在使用该方法前，Parameter 对象的 Attributes 参数必须设置为 adFLDLong，这样 Parameter 对象能够接收该方法加入的长文本或二进制数据。当多次调用该方法时，新数据就可以连续性地加入到现存的参数中。

9.4 习　题

9.4.1 填空题

(1) 数据库本身由多个数据表所组成，表中每一行代表一个____________，每一列代表一个____________。

(2) 索引本身根据其功用可以分为两种，____________和____________。

(3) ____________指定 SELECT 语句查询及与查询相关的表或视图。

(4) 建立 Connection 对象是采用 Server 对象的____________方法进行的。

9.4.2 选择题

(1) 在 ODBC 数据源管理器中，DSN 还分为用户 DSN、系统 DSN 和文件 DSN 这 3 种。用户可以通过 ODBC 数据源管理器创建(　　)类型的 DSN。

A. 1 种　　B. 2 种

C .3 种　　D. 以上全错

(2) 在开发基于数据库的 Web 应用程序时，构建(　　)都是可以的

A. 系统 DSN 和文件 DSN　　B. 文件 DSN 和用户 DSN

C. 系统 DSN 和用户 DSN　　D. 以上全错

9.4.3 问答题

(1) OLE DB 有哪两层结构，有什么优势？

(2) 在 ADO 模型中有哪些对象？

(3) 简述 Command 对象和 Connection 对象之间的关系。

9.4.4 操作题

(1) 在 Access 2003 中创建一个名为 CDB.mdb 的数据库，并在该数据库中设置一个如图 9-30 所示的数据表 Custom。

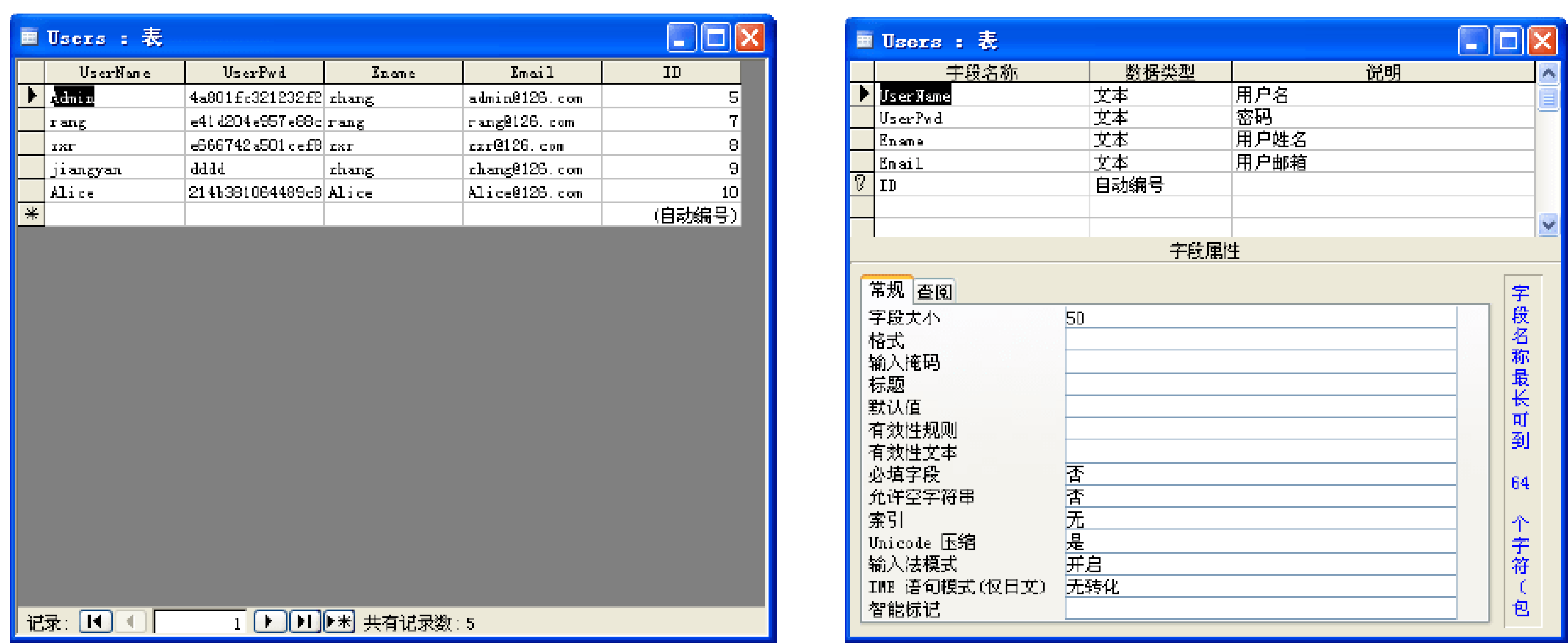


图 9-30　创建数据表

(2) 以上例创建的数据库为基础，参考本章练习所介绍的方法，创建数据库连接。

第10章　RecordSet 对 象

RecordSet 对象是一个记录的集合，是按字段(或列)和记录(或行)的形式构成的二维表。每个 RecordSet 对象表示表中的记录或运行一次查询所得到的结果。使用 RecordSet 对象，可以在记录一级上对数据库中的数据进行各种操作，如增、删和定位记录，以及更新数据库等。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 了解用 Recordset 对象处理结果的方法
- 掌握 Recordset 的应用
- 了解 Fields 集合和 Field 对象

10.1　认识 Recordset 对象

对于检索数据，检查结果，更改数据库，ADO 提供了 Recordset 对象。Recordset 对象是一个记录的集合，用于检索和更新数据库。数据库应用程序通常用 Connection 对象建立连接并用 Recordset 对象处理返回的数据。

尽管 Connection 对象简化了连接数据库和查询任务，但 Connection 对象仍有许多不足：检索和显示数据库信息的 Connection 对象不能用于创建脚本；用户必须确切地知道要对数据库作出的更改，然后才能使用查询实现更改。

10.1.1　Recordset 对象简介

Recordset 对象可以创建一个记录集合，并且将所需的记录从表中取出。同时，使用虚拟表格的方式，每一行为一条记录，每一列则代表一个字段，提供给 ASP 程序处理，如图 10-1 所示。

	分类编号	产品代号	产品名称	产品简述
	55	0001000001	LA-Gear Mouse PAD	MODEL No.TMP– 0312
记录指针 ⇨	55	0001000002	LA-Gear Mouse PAD 1	MODEL No.TMP– 04

图 10-1　Recordset 对象的记录指针

- Recordset 中的记录指针具有游标类型(CursorType)。不同的游标类型可对记录集进行不同的操作，默认值为 0，代表记录指针只能向前移动记录集，也可定义成其他值，允许记录指针在记录集中上下移动。

- 数据源本身具有锁定的能力。具有这项功能最主要的目的在于避免两个 SQL 查询操作同时写同一条记录。
- 当前记录指针的位置。Recordset 的 MoveFirst 方法可以将记录指针移到第一条记录的位置，MoveLast 方法可将记录指针移到记录集合的最后一条，MoveNext 方法可使指针移到下一条，MovePrevious 方法则是移到上一条。

1. Recordset 对象的属性

Recordset 对象的属性及其相关说明如表 10-1 所示。

表 10-1　Recordset 对象的属性

属　性	说　明
AbsolutePage	设置或返回当前记录所在的页码。用该属性可使当前记录跳到指定的页，如：<% rs.AbsolutePage = 6 %>
AbsolutePosition	设置或返回当前记录在记录集中的位置。用该属性可使某一记录成为当前记录，如：<% rs.AbsolutePosition = 10 %>
ActiveConnection	定义 Recordset 对象与数据库的连接。该属性或者指向一个当前打开的 Connection 对象，或者定义一个新的连接
BOF	若记录指针位于第一条记录之前，则为 True，否则为 False
BookMark	设置或返回一个记录的书签，如 mark=rs.BookMark 或 rs.BookMark=mark
CacheSize	设置本地计算机可以缓存的记录数，默认值为 1
CursorType	设置记录集所用的游标类型，详见表 10-3
CursorLocation	设置游标位置，若设成 1 或 adUseClient，使用客户端提供的本地游标；若设成 2 或 adUseServer(默认值)，则使用数据提供者的游标
EditMode	指出当前记录的编辑状态，取值如下。 ● 0 或常量 adEditNone：在当前处理过程中没有编辑操作 ● 1 或常量 adEditInProgress：当前记录已被更改，但尚未保存到数据库 ● 2 或常量 adEditAdd：当前缓冲区内数据是用 AddNew 方法写入的新记录，但尚未保存到数据库 ● 3 或常量 adEditDelete：当前记录已被删除
EOF	若记录指针位于最后一条记录之后，则为 True，否则为 False
Filter	定义筛选器来获取特定的记录，如 rs.Filter="AuthorID">1000
LockType	设置记录集所用的锁定类型，详见表 10-4
MaxRecords	确定一次所能返回的最大记录数，默认值为 0，表示返回全部请求的记录。在记录集关闭时该属性可读写，在打开记录集后只读
PageCount	记录集所包含的页数，每页记录数由 PageSize 决定
PageSize	指定一页中包含的记录数

(续表)

属　性	说　明
RecordCount	记录集所包含的记录条数
Sort	设置记录集的排序方式
Source	设置记录集数据来源，可以是 Command 对象、SQL 语句、表名或存储过程
State	确定记录集的打开/关闭状态，取值如下。 ● 0 或 adStateClosed：表示记录集已关闭 ● 1 或 adStateOpen：表示记录集已打开 ● 2 或 adStateConnecting：表示正在连接 ● 3 或 adStateExecuting：表示记录集正在执行一个命令 ● 4 或 adStateFetching：表示记录集正在获取数据
Status	当前记录所处的状态，详见表 10-5

【练习 10-1】创建一个搜索页面，当用户在表单中输入出版社名称并单击“提交”按钮后，可返回数据库中该出版社所有图书的详细信息，程序运行效果如图 10-2 和图 10-3 所示(在操作本实例前，用户应结合本书第 9 章所介绍的内容创建搜索页面的数据库并输入数据)。

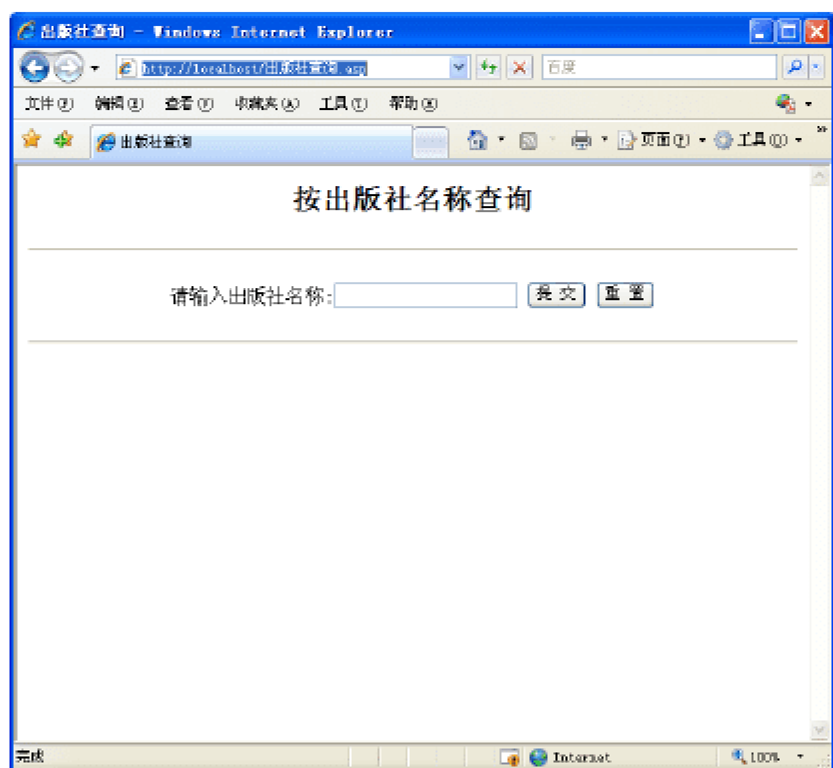
(1) 创建一个名为“出版社查询.asp”的网页，其代码如下所示：

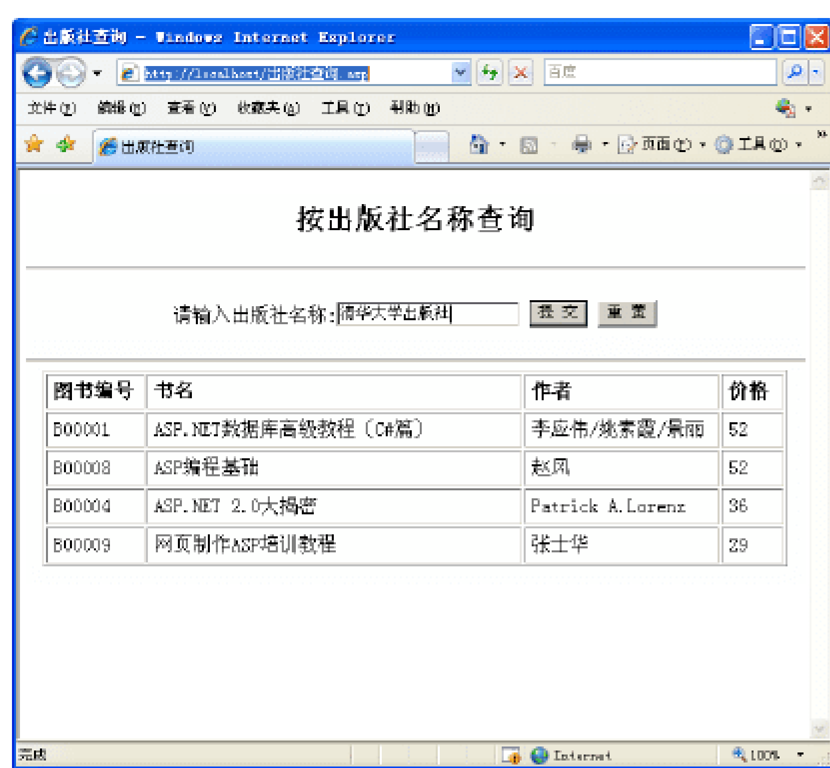
```
<%@ Language=VBScript %>
<% Response.Buffer = True %>
<HTML>
<TITLE>出版社查询</TITLE>
<HEAD>
<SCRIPT Language="VBScript" RUNAT="Server">
Sub rs_Display()
    dim strConn,strSQL,strWriter
    strPress = Request.Form("PressName")
if strPress<>"" then
    '创建 Connection 对象 conn
    Set conn=Server.CreateObject("ADODB.Connection")
    '使用 DSN 建立 conn 与数据库的连接
    strConn="Driver={SQL Server};Database=booknet;Server=ZHANGSHIHUA\TEST;UID=sa;
      PWD=;"
    conn.Open strConn
    strSQL = "SELECT * FROM Books"
    strSQL = strSQL & " WHERE Press LIKE '%" & Trim(strPress) & "%'"
    Set rs = Server.CreateObject("ADODB.Recordset")
    rs.Open strSQL,conn
    With Response
      if rs.EOF then
```

```
                .Write "没有查询到相关的记录！"
            else
                    .Write "<TABLE BORDER=1 CELLSPACE=0 CELLPADDING=5>" &_
                        "<TR HEIGHT=12><TD WIDTH=70><B> 图书编号 </B></TD>" &_
                        "<TD WIDTH=300><B> 书名 </B></TD>" &_
                        "<TD WIDTH=150><B> 作者 </B></TD>" &_
                        "<TD WIDTH=40><B> 价格 </B></TD></TR>"
            end if
            do Until rs.EOF
                    .Write "<TR HEIGHT=12><TD WIDTH=70>" & rs("BookID") & "</TD>" &_
                        "<TD WIDTH=300>" & rs("BookName") & "</TD>" &_
                        "<TD WIDTH=150>" & rs("Writer") & "</TD>" &_
                        "<TD WIDTH=40>" & rs("Price") & "</TD></TR>"
            rs.MoveNext
            loop
            .Write "</TABLE>"
        End With
            rs.Close
        conn.Close
end if
end sub
</SCRIPT>
<BODY>
<Center><H2>按出版社名称查询</H2><HR>
<FORM NAME="thisForm" METHOD=POST
            ACTION="<%=Request.ServerVariables("Script_Name")%>" >
        <P>请输入出版社名称:<INPUT TYPE="text" NAME="PressName" SIZE=20>
        <INPUT TYPE="submit" NAME="btnSubmit" value="提  交">
        <INPUT TYPE="reset" NAME="btnReset" value="重  置"></P>
</FORM>
<Hr>
<%
        call rs_Display()
%>
</Center></BODY></HTML>
```

(2) 启动 IE 浏览器后，在地址栏中输入 http://localhost/出版社查询.asp 并按 Enter 键，效果如图 10-2 所示。

(3) 在图 10-2 所示网页中的文本框内输入“清华大学出版社”后，单击“提交”按钮，网页效果如图 10-3 所示。


图 10-2　查询页面效果


图 10-3　网页查询效果

注意：

用户操作本章实例的过程中可以参考本书第 9 章所介绍的内容，根据自己的实际情况设置程序代码中的数据库连接部分。

2. Recordset 对象的方法

Recordset 对象的方法及其相关说明如表 10-2 所示。

表 10-2　Recordset 对象的方法

方　法	说　明
AddNew	增加一条记录
CancelBatch	取消一个批处理更新
CancelUpdate	在更新前取消对当前的所有更改
Clone	建立记录集的一个副本
Delete	删除一条或多条记录
GetRows	从记录集中得到多条记录并存入数组中
Move	将记录指针移到指定的位置
MoveFirst	将记录指针移到第一条记录处
MoveLast	将记录指针移到最后一条记录处
MoveNext	将记录指针移到下一条记录处
MovePrevious	将记录指针移到前一条记录处
NextRecordSet	从能产生多个结果的命令中返回下一个记录集
Open	打开记录集
Requery	重新执行查询来刷新记录集
Resync	刷新服务器内的同步数据
Save	将记录集保存到一个文件中

(续表)

方　　法	说　　明
Supports	判断记录集是否支持指定的功能
Update	将修改结果保存到数据库中
UpdateBatch	将缓冲区内批量修改结果保存到数据库中

10.1.2　Recordset 对象的工作流程

在使用 Recordset 对象前同样需要使用 Connection 对象建立数据库的连接，其步骤如下：

(1) 创建 Connection 对象，打开数据源。创建一个 Connection 对象，并保存在 OBJConn 变量中，然后打开数据源，程序代码如下：

```
Set OBJConn = Server.CreateObject("ADODB.Connection")
OBJConn.Open strDSN  ' 打开数据源
```

(2) 创建 Recordset 对象。在取得与数据库的连接后，即可创建 Recordset 对象，其程序代码如下：

```
Set Rs = OBJConn.Execute( SQLstr )
```

(3) 打开 Recordset 对象取得数据。在 Recordset 对象创建完成后，即可打开 Recordset 对象的内容。此 Recordset 对象的内容可以是表、SQL 查询语句，如果是表，其命令如下所示：

```
Rs.Open "产品基本信息", OBJConn, adOpenStatic, adLockReadOnly, adCmdTable
```

或是：

```
Set Rs = OBJConn.Exceute(CommandText, RecordsAffected, Options)
```

上述程序代码使用 Recordset 对象 Rs 的 Open 方法打开“产品基本信息”表，第二个参数为 Connection 对象 OBJConn，其后三个参数为定义在文件 adovbs.inc 中的常数，用来定义 Recordset 对象的存取方式。

(4) 处理 Recordset 对象的记录。在打开 Recordset 对象的记录集合后，即可开始使用 Recordset 对象的属性及方法进行表的操作或取得当前的状态。例如取得 Recordset 对象的状态属性 State，代码如下：

```
If Rs.State = 1 Then
    Response.Write("<b>Rs 对象目前处于打开的状态</b><br>")
Else
    Response.Write("<b>Rs 对象目前处于关闭的状态</b><br>")
End If
```

上述程序代码利用 If...Then...Else 语句的方式检查状态属性 State，查看当前 Recordset 对象的打开状态。如果返回值为 1，代表状态已打开；返回值为 0，则代表状态已关闭。

(5) 关闭 Recordset 对象。关闭 Recordset 对象的代码如下：

```
Rs.Close
Set Rs = Nothing
```

在执行上述程序代码后，将会关闭 Recordset 对象，并且由 Set Rs = Nothing 释放 Recordset 对象。

(6) 关闭与数据库的连接。要关闭数据库连接，可使用 Connection 对象的 Close 方法，代码如下：

```
OBJCConn.Close
set OBJConn = Nothing
```

10.2　应用 Recordset 对象

Recordset 对象可以根据查询条件，检索并显示一组数据库记录。Recordset 对象保持查询返回的记录位置，允许一次一项逐步扫描结果，根据 Recordset 对象的指针类型属性设置，可以滚动显示和更新记录。指针可以在一组记录中定位到特定的项，还用于检索和检查记录，然后在这些记录的基础上执行操作。

10.2.1　游标类型

游标类型(CursorType)代表不同的数据获取方法。打开记录集时，可在 Open 方法中指定 Recordset 对象所用的游标类型，或在调用 Open 方法前用 CursorType 属性来设置游标类型。记录集打开后，CursorType 属性是只读的，可以用该属性来返回游标类型。

游标类型的取值与相关说明如表 10-3 所示。

表 10-3　游标类型(CursorType)

类　型	常　量　名	值	说　　明
仅向前	AdOpenForwardOnly	1	只能向前浏览记录。对简单的浏览可提高性能，但很多属性和方法(如：BookMark、RecordCount、AbsolutePage、AbsolutePosition 等)不能使用
键集	adOpenKeyset	2	其他用户对记录所做的修改将反映到记录集中，但其他用户增加或删除的记录不会反映到记录集中。键集游标支持 BookMark 属性，支持全功能浏览，可以使用 RecordCount、AbsolutePage 和 AbsolutePosition 等属性

(续表)

类　型	常　量　名	值	说　　明
动态	adOpenDynamic	3	动态游标功能最强，但消耗资源也最多。使用动态游标时，其他用户对记录所做的增加、删除或修改的记录都会反映到记录集中。动态游标支持全功能浏览
静态	adOpenStatic	4	静态游标只是数据的一个快照，其他用户对记录所做的增加、删除或修改的记录都无法反映到记录集中。静态游标支持向前或向后移动

一旦打开 RecordSet 对象，就不能改变 CursorType 属性。但是关闭 Recordset(采用关闭方式)对象后可以改变 CursorType 属性，然后重新打开 RecordSet 对象，那么就可以有效地改变它的类型。

注意：

用户可依据需求，指定 CursorType 为上述指针中的任何一种，如省略则取其默认值 adOpenForwardOnly。这是功能最少的记录集，耗费的资源也最少。

以上几个游标类型将直接影响到 Recordset 对象所有的属性和方法，当显示一个表时，不同的指针类型将会影响到这个表的属性和方法，如表 10-4 所示。

表 10-4　游标类型对 Recordset 属性的影响

Recordset 属性	adOpenForwardOnly	adOpenKeyset	adOpenDynamic	adOpenStatic
AbsolutePage	不支持	不支持	可读写	可读写
AbsolutePosition	不支持	不支持	可读写	可读写
BOF	只读	只读	只读	只读
CursorType	可读写	可读写	可读写	可读写
EOF	只读	只读	只读	只读
Filter	可读写	可读写	可读写	可读写
LockType	可读写	可读写	可读写	可读写
PageCount	不支持	不支持	只读	只读
PageSize	可读写	可读写	可读写	可读写
RecordCount	不支持	不支持	只读	只读
AddNew	支持	支持	支持	支持
CancelBatch	支持	支持	支持	支持
CancelUpdate	支持	支持	支持	支持
Close	支持	支持	支持	支持
Delete	支持	支持	支持	支持

(续表)

Recordset 属性	adOpenForwardOnly	adOpenKeyset	adOpenDynamic	adOpenStatic
Move	不支持	支持	支持	支持
MoveFirst	支持	支持	支持	支持
MoveLast	不支持	支持	支持	支持
MoveNext	支持	支持	支持	支持
MovePrevious	不支持	支持	支持	支持
Open	支持	支持	支持	支持
Update	支持	支持	支持	支持
UpdateBatch	支持	支持	支持	支持

【练习 10-2】创建一个能进行数据搜索和操作的页面，能根据图书的名称在数据库中查询符合条件的图书，还可以链接到相应的页面对该记录进行编辑和删除操作，程序运行效果如图 10-4 所示(在操作本实例前，用户应结合本书第 9 章所介绍的内容创建搜索页面的数据库并输入数据)。

(1) 创建一个名为“超链查询.asp”的网页，其代码如下所示：

```
<%@ Language=VBScript %>
<% Response.Buffer = True %>
<HTML>
<TITLE>图书查询</TITLE>
<HEAD>
<SCRIPT Language="VBScript" RUNAT="Server">
Sub rs_Display()
    dim strConn,strSQL,strBookName
    strBookName = Request.Form("BookName")
    Set conn=Server.CreateObject("ADODB.Connection")
    strConn="Driver={SQL Server};Database=booknet;&-
Server=ZHANGSHIHUA\TEST;UID=sa;PWD=;"
    conn.Open strConn
    strSQL = "SELECT * FROM Books"
    strSQL = strSQL & " WHERE BookName LIKE '%" & Trim(strBookName) & "%'"
    strSQL = strSQL & " ORDER BY BookName"
    Set rs = Server.CreateObject("ADODB.Recordset")
    rs.Open strSQL,conn
    With Response
      if rs.EOF then
         .Write "没有查询到相关的记录！"
      else
```

```
            .Write "<TABLE BORDER=1 CELLSPACE=0 CELLPADDING=5>" &_
                "<TR HEIGHT=12><TD WIDTH=20><B> BookID </B></TD>" &_
                "<TD WIDTH=270><B> 书名 </B></TD>" &_
                "<TD WIDTH=200><B> 作者 </B></TD>" &_
                "<TD WIDTH=40><B> 价格 </B></TD>" &_
                "<TD WIDTH=40><B> 修改 </B></TD>" &_
                "<TD WIDTH=40><B> 删除 </B></TD></TR>"
        end if
        do Until rs.EOF
            .Write "<TR HEIGHT=12><TD WIDTH=20>" & rs("BookID") & "</TD>" &_
                "<TD WIDTH=270>" & rs("BookName") & "</TD>" &_
                "<TD WIDTH=200>" & rs("Writer") & "</TD>" &_
                "<TD WIDTH=40>" & rs("Price") & "</TD>" &_
                "<TD WIDTH=40><A HREF=" & chr(34) & "10-5-编辑页面.asp?BookID='" &_
                        rs("BookID") & "'" & chr(34) & ">" & "编辑" & "</A></TD>" &_
                  "<TD WIDTH=40><A HREF=" & chr(34) & "10-6-删除.asp?BookID='" &_
                rs("BookID") & "'" & chr(34) & ">" & "删除" & "</A></TD>" &_
                "</TR>"
            rs.MoveNext
        loop
        .Write "</TABLE>"
    End With
    rs.Close
    conn.Close
end sub
</SCRIPT>
<BODY>
<Center><H4>带超链接的图书查询页面</H4>
<Hr>
<FORM NAME="thisForm" METHOD=POST
        ACTION="<%=Request.ServerVariables("Script_Name")%>" >
  <P>请输入书名:<INPUT TYPE="text" NAME="BookName" SIZE=20>
  <INPUT TYPE="submit" NAME="btnSubmit" value=" 查 找 ">
  <INPUT TYPE="reset" NAME="btnReset" value=" 重 置 ">
  <INPUT TYPE="button" NAME="btnInsert" value="插 入"&-
        onclick="location.href='10-7-插入页面.htm'">
  </P>
</FORM>
<%
  call rs_Display()
%>
<HR>
</Center></BODY></HTML>
```

(2) 运行“超链查询.asp”的网页页面后，用户不仅能根据图书的名称在数据库中搜索符合条件的图书，还可以链接到相应的页面对页面记录进行编辑和删除操作，如图 10-4 所示。



图 10-4　图书查询页面

10.2.2　锁定类型

锁定类型(LockType)是针对数据库操作中并发事件的发生而提出的系统安全控制方式。锁定类型不仅影响 Recordset 对象的并发事件的控制处理方式，而且决定了记录集是否能更新以及记录集的更新是否能批量地进行。

打开记录集时，可以在 Open 方法中指定锁定类型，或者在调用 Open 方法前用 LockType 属性来设置锁定类型。锁定类型的取值与相关说明如表 10-5 所示。

表 10-5　锁定类型(LockType)

类　型	常　量　名	值	说　　明
只　读	adLockReadOnly	—	以只读方式打开记录集时，不能改变任何数据，只读方式是默认的锁定方法
保守式	adLockPessimistic	2	当编辑时立即锁定记录，这是最安全的锁定方法
开放式	adLockOptimistic	3	数据提供者只有在调用 Update 方法时才锁定记录，而在此之前其他操作者仍可对当前记录进行增加、删除或修改等操作
开放式批处理	adLockBatchOptimistic	4	当编辑记录时记录不会被锁定，而增加、删除或修改记录是在批处理方式下完成的

如果数据源没有返回记录，那么提供者将 BOF 和 EOF 属性同时设置为 True，并且不定义当前记录位置。如果游标类型允许，仍然可以将新数据添加到该空 Recordset 对象。

注意：

锁定类型的设定会影响数据的修改程序，若是没有指定锁定类型则将会返回一个默认只读的记录集对象，其中的数据将无法修改。

10.2.3　浏览记录

记录集对象成功引用 Open 方法后，SQL 语句所取得的数据副本会存储在记录集对象之中，此时的记录集对象类似一个包含特定数据的原始数据表，例如以下的程序片段：

```
ObjRst.Open "select * from Customers ","dsn=test;"
```

这段程序代码返回的记录集对象 ObjRst，实际上就是 Customers 数据表的内容，可以通过移动记录集对象的指针，浏览数据表中每一条记录的特定字段内容。

指针总是指向记录集当前的数据位置，Recordset 对象提供 4 个重要的数据浏览方法：MoveFirst、MovePrevious、MoveNext 和 MoveLast，分别让将当前指针移到记录集的首记录、前一个记录、后一个记录和末记录，如图 10-5 所示。

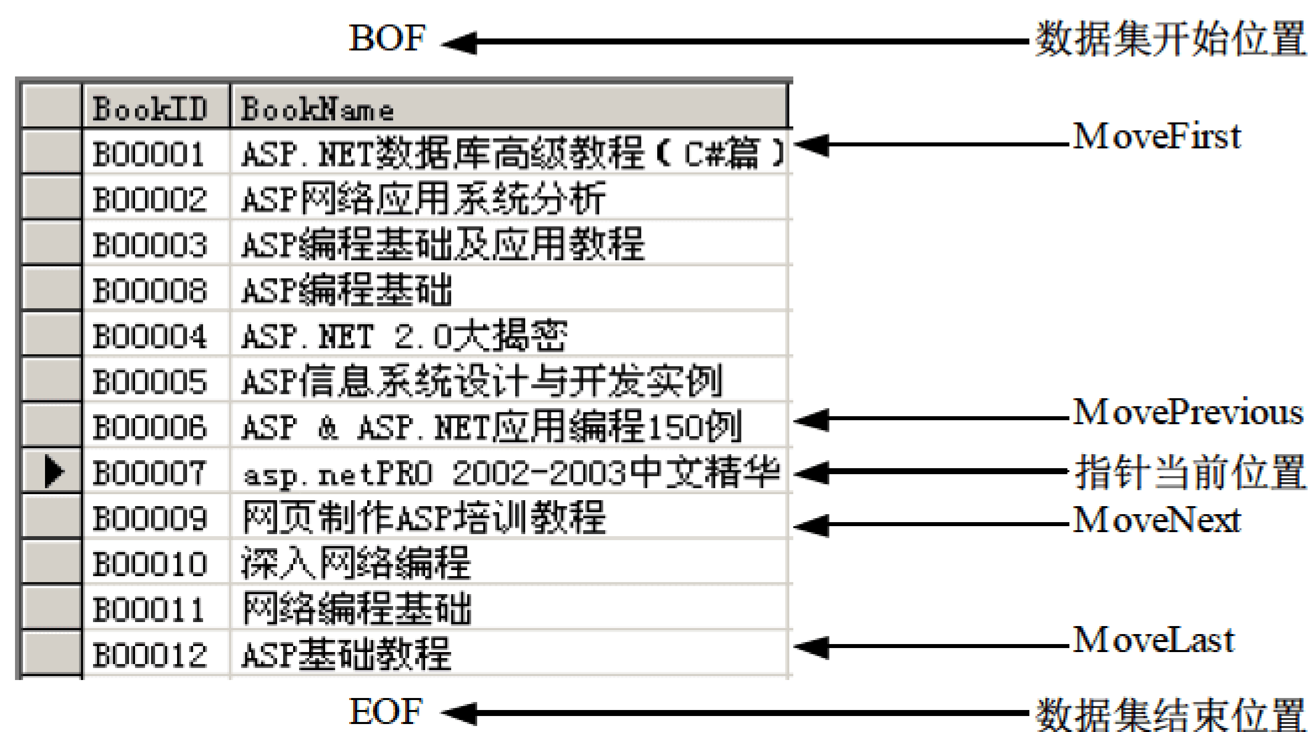


图 10-5　指针的移动

当指针位于记录集的结束位置，则函数 EOF 的返回值为 True，否则为 Flase。对于查询返回的记录集对象，其默认的指针位置为首记录。通过指定数据对象的字段名称，可以取得记录集当前指针位置所在的这条记录中的字段内容。

记录集在浏览记录时会受到其打开时所指定的游标类型的限制。例如，一个设定为 adOpenForwardOnly 的记录集，只能往前移动，因此引用 MoveFirst 或 MovePrevious 将会引发不允许操作的错误。而 adOpenDynamic 则拥有最大的自由度，可以随意引用任何一个方法浏览记录集。

如图 10-5 所示，有两个标示为 BOF 和 EOF 的特殊指针位置，分别代表记录集开始与结束的位置，这两个位置只是一个空值。当指针移动至首记录时，使用 MovePrevious 可以达到 BOF 的位置，反之引用 MoveLast，可以将指标移到 EOF。

记录集对象提供了 BOF 和 EOF 属性，用来判断当前指针是否位于记录集的开始或结束的位置，因此在应用程序中，通常都会通过查看记录集对象的 BOF 或 EOF 属性值是否

为 True，来了解指针是否位于记录集的开头或结束位置，代码如下：

```
blnEof = objRst.EOF
```

10.2.4　添加记录

添加记录有两种方法：一种是用 SQL INSERT INTO 语句，另一种是用 Recordset 对象的 AddNew 方法。

如果希望在数据库特定表中添加一条记录，则必须使用 INSERT 语句，INSERT 语句的语法结构如下：

```
INSERT INTO 表名 [(字段名称 1 [, 字段名称 2[, ...]])]
VALUES (数据 1 [, 数据 2 [, ...]])
```

在使用 INSERT 添加数据时必须注意以下几点：

- 表名后括号内的字段名称的排列顺序与 VALUES 后括号内数据的排列顺序必须一致，否则进行数据添加时会失败。
- 对未设置的字段，添加数据时其默认值为 Null。
- 对于新添加的数据记录，在建立时就给予其默认值 Null。

【练习 10-3】打开一个 Recordset 对象，然后用 AddNew 方法插入一条空记录，填充空记录的各个字段，再调用 Update 方法把记录写到数据库中，程序运行效果如图 10-6 所示。

(1) 确定 adovbs.inc 文件在应用程序的当前目录中。

(2) 建立显示添加记录内容的“添加记录.html”文件。

```
<html><head><title>添加一条新记录</title></head>
<body>
<Center><H4>添加一条新记录</H4></Center>
<Hr>
<FORM Method=POST Action="添加记录.asp" >
<table border="0" width="100%">
  <tr><td align="right" width="20%"><B>书名：</B></td>
    <td width="80%"><input type="text" name="BookName" size=50>
<Font color=Red> **必须输入**</Color></td></tr>
  <tr><td align="right"><b>作者：</b></td>
    <td><input type="text" name="Writer" size=30></td></tr>
  <tr><td align="right"><b>出版社：</b></td>
    <td><input type="text" name="Press" size=50></td></tr>
  <tr><td align="right"><b>出版日期：</b></td>
    <td><input type="text" name="PublishDate" size=10></td></tr>
  <tr><td align="right"><b>定价：</b></td>
    <td><input type="text" name="Price" size=10></td></tr>
  <tr><td align="right"><b>内容简介：</b></td>
```

```
        <td> <TEXTAREA COLS=50 ROWS=4 name="Statement"></TEXTAREA></td></tr>
    <tr><td align="right"><b>类别：</b></td>
        <td><input type="text" name="category" size=10></td></tr>
    <tr><td align="right"><b>子类：</b></td>
        <td><input type="text" name="SubCategory" size=10></td></tr>
    <tr><td align="right"><b>库存量：</b></td>
        <td><input type="text" name="Quantity" size=10></td></tr>
<tr><td COLSPAN=2 ALIGN=Center>
            <input type="submit" value=" 保 存 ">
        <input type="reset" value=" 重 置 ">
        <input type="button" value=" 返 回 " onclick="location.href='index.asp'"></td></tr>
  </table>
</FORM>
</body></html>
```

(3) 建立处理添加记录内容的“添加记录.asp” 文件。

```
<%@ Language=VBScript %>
<!--#include file="adovbs.inc"-->
<%
if IsNumeric(Request.Form("Quantity")) and _
  IsNumeric(Request.Form("Price")) and _
    IsDate(Request.Form("PublishDate")) then
              strConn="Driver={SQL Server};Database=booknet;&-
Server=ZHANGSHIHUA\TEST;UID=sa;PWD=;"
        Set rs = Server.CreateObject("ADODB.Recordset")
rs.CursorType = adOpenKeyset
        rs.LockType = adLockOptimistic
        rs.Open "Books",strConn,,,adCmdTable
        rs.AddNew                    '添加一条新记录
        rs("BookID") = "B" & Zeros(5-len(rs.RecordCount + 1)) & CStr((rs.RecordCount + 1))
        rs("BookName") = Trim(Request.Form("BookName"))
        rs("Writer") = Trim(Request.Form("Writer"))
        rs("Press") = Trim(Request.Form("Press"))
        rs("PublishDate") = CDate(Trim(Request.Form("PublishDate")))
        rs("Statement") = Trim(Request.Form("Statement"))
        rs("Category") = Trim(Request.Form("Category"))
        rs("SubCategory") = Trim(Request.Form("SubCategory"))
        rs("Quantity") = CInt(Trim(Request.Form("Quantity")))
        rs("Price") = CCur(Trim(Request.Form("Price")))
        if rs.Supports(adUpdate) then
            rs.Update
         Response.Redirect("添加记录.html")
        end if
        rs.Close
```

```
else
    Response.Write "输入的数量、价格、出版日期中有错，请重新输入！"
end if
%>
<SCRIPT Language="VBScript" RUNAT="Server">
Function Zeros(intNum)
    Dim I,strOut
    for I = 1 to intNum
        strOut = strOut & "0"
    next
    Zeros = strOut
End Function
</SCRIPT>
```

(4) 运行程序后，效果如图 10-6 所示。用户可以在该页面中向数据库中添加记录。

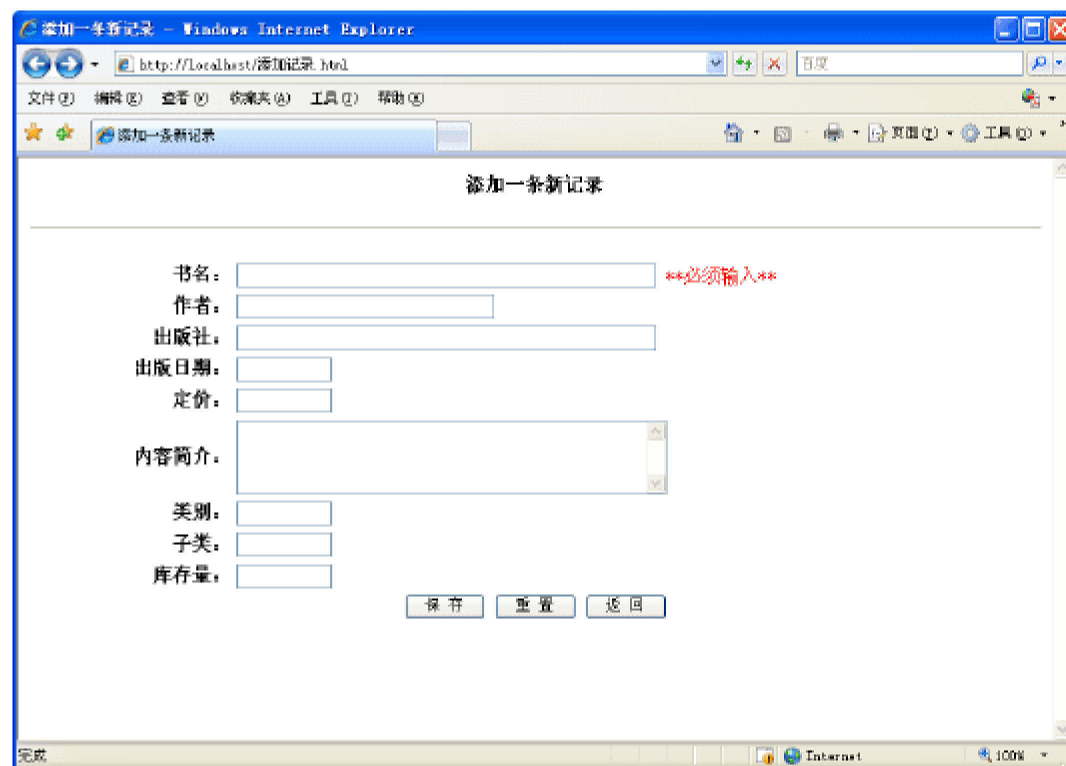


图 10-6　添加新记录

10.2.5　更新记录

有两种方法可对记录进行更新，一种是用 SQL UPDATE 语句修改记录，另一种是用 Recordset 对象的 Update 方法修改记录。

要更新数据库中的记录，游标类型应设置成动态，更新数据使用 UPDATE 语句，其基本语法如下：

```
UPDATE 表名
SET 字段 1=数据 1，字段 2=数据 2,…
WHERE 筛选条件
```

更新操作有以下 3 种情况。

- 所有记录全部更新：可以对表中所有的记录同时进行内容的更新。
- 不更新索引键值而只更新其他字段的内容：仅仅更新每一条记录内的非索引字段内容。
- 更新索引键值：必须连带地更新其他表中相关联的记录内容。

【练习 10-4】打开一个记录集，然后更改其数据，再调用 Update 或 UpdateBatch 方法把所作的更改写入到数据库中。程序运行效果如图 10-7 所示。

(1) 确定 adovbs.inc 文件在应用程序的当前目录中。

(2) 建立显示和处理修改记录内容的“更新记录.asp”文件，其代码如下所示：

```
<%@ Language=VBScript %>
<!--#include file="adovbs.inc"-->
<% Response.Buffer = True %>
<SCRIPT Language="VBScript" RUNAT="Server">
Dim rs
Dim conn
Dim strBookID
sub Get_Record()
      strBookID = Request.QueryString("BookID")
'创建 Connection 对象 conn
Set conn=Server.CreateObject("ADODB.Connection")
'使用 DSN 建立 conn 与数据库的连接
strConn="Driver={SQL
          Server};Database=booknet;Server=ZHANGSHIHUA\TEST;UID=sa;PWD=;"
conn.Open strConn
     strSQL = "SELECT * FROM Books "
     strSQL = strSQL&" WHERE BookID =" & Trim(strBookID)
 Set rs = Server.CreateObject("ADODB.Recordset")
rs.Open    strSQL,conn
End sub
sub Update_Record()
     if IsNumeric(Request.Form("Quantity")) and _
          IsNumeric(Request.Form("Price")) and _
          IsDate(Request.Form("PublishDate")) then
            Set rs = Server.CreateObject("ADODB.Recordset")
            rs.CursorType = adOpenStatic
            rs.LockType = adLockOptimistic
            strConn="Driver={SQL
                      Server};Database=booknet;Server=ZHANGSHIHUA\TEST;UID=sa;PWD=;"
            strSQL = "SELECT * FROM Books"
            strSQL = strSQL & " WHERE BookID =" & Trim(strBookID)
            rs.Open strSQL,strConn,,,adCmdText
            if rs.EOF then
                  Response.Write "没有查询到相关的记录！"
                  Exit sub
            end if
            if rs.Supports(adUpdate) then
                  rs("BookName") = Trim(Request.Form("BookName"))
```

```
                rs("Writer") = Trim(Request.Form("Writer"))
                rs("Press") = Trim(Request.Form("Press"))
                rs("PublishDate") = CDate(Trim(Request.Form("PublishDate")))
                rs("Statement") = Trim(Request.Form("Statement"))
                rs("Category") = Trim(Request.Form("Category"))
                rs("SubCategory") = Trim(Request.Form("SubCategory"))
                rs("Quantity") = CInt(Trim(Request.Form("Quantity")))
                rs("Price") = CCur(Trim(Request.Form("Price")))
                rs.Update
                if conn.Errors.Count > 0 then
                    Response.Write "Transacton Error"
                else
                    Response.Write "Transacton Ok"
                end if
                Response.Redirect("index.asp")
            else
                Response.Write "没有能够更新数据！"
            end if
            rs.Close
      else
            Response.Write "输入的数量、价格、出版日期中有错，请重新输入！"
      end if
      rs.Close
      conn.Close
      Set rs = Nothing
      Set conn = Nothing
end sub
Function Zeros(intNum)
    Dim I,strOut
    for I = 1 to intNum
        strOut = strOut & "0"
    next
    Zeros = strOut
End Function
</SCRIPT>
<html><head><title>修改一条现有记录</title></head>
<body>
<Center><H4>修改一条现有记录</H4></Center>
<Hr>
<% call Get_Record() %>
<% if Request.Form("btnUpdate") = " 保 存 " then call Update_Record() %>
<FORM action="<%=Request.ServerVariables("Script_Name")%>" Method=POST>
<table border="0" width="100%">
```

```
    <tr><td align="right" width="20%"><B>书名：</B></td>
      <td width="80%"><input type="text" name="BookName" &-
            size=50 value=<%= rs("BookName")%>>
<Font color=Red> **必须输入**</font></td></tr>
    <tr><td align="right"><b>作者：</b></td>
      <td><input type="text" name="Writer" size=30 value=<%=rs("Writer")%>></td></tr>
    <tr><td align="right"><b>出版社：</b></td>
      <td><input type="text" name="Press" size=50 value=<%=rs("Press")%>></td></tr>
    <tr><td align="right"><b>出版日期：</b></td>
      <td><input type="text" name="PublishDate" size=10
               value=<%=rs("PublishDate")%>></td></tr>
    <tr><td align="right"><b>内容简介：</b></td>
      <td><TEXTAREA COLS=50 ROWS=4
               name="Statement"><%=rs("Statement")%></TEXTAREA></td></tr>
    <tr><td align="right"><b>类别：</b></td>
      <td><input type="text" name="category" size=10 value=<%=rs("Category")%>></td></tr>
    <tr><td align="right"><b>子类：</b></td>
      <td><input type="text" name="SubCategory" size=10
               value=<%=rs("SubCategory")%>></td></tr>
    <tr><td align="right"><b>库存量：</b></td>
      <td><input type="text" name="Quantity" size=10 value=<%=rs("Quantity")%>></td></tr>
    <tr><td align="right"><b>定价：</b></td>
      <td><input type="text" name="Price" size=10 value=<%=rs("Price")%>></td></tr>
    <tr><td COLSPAN=2 ALIGN=Center>
        <input type="submit" name="btnUpdate" value=" 保 存 ">
        <input type="button" value=" 返 回 " onclick="location.href='index.asp'"></td></tr>
   </table>
</FORM>
</body></html>
```

(3) 运行 index.asp 文件，单击某记录后的“编辑”超链接以执行记录的编辑操作，如图 10-7 所示。

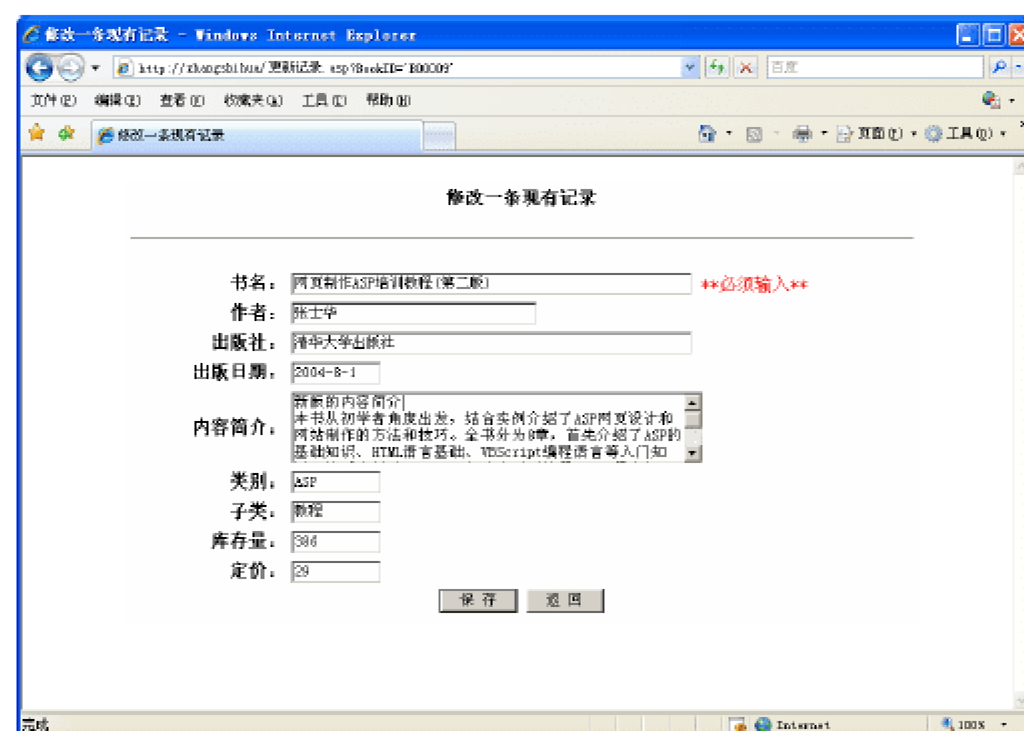


图 10-7　修改现有记录

10.2.6　删除记录

有两种方法可对记录进行更新：一种是用 SQL DELETE 语句删除记录，另一种是用 Recordset 对象的 Delete 方法删除记录。

使用 DELETE 语句时在表中删除一条记录，必须特别注意，一旦数据被删除，将无法恢复。如果没有指定 WHERE 子句，将删除表中的所有记录。以下是 DELETE 语句的语法结构：

```
DELETE FROM 表名 WHERE 筛选条件
```

【练习 10-5】打开一个记录集，然后用 Delete 方法删除其中的记录，程序运行效果如图 10-8 所示。

(1) 创建删除记录页面的“删除记录.asp”文件，其代码如下所示：

```
<%
strBookID = Request.QueryString("BookID")
Set conn=Server.CreateObject("ADODB.Connection")
strConn="Driver={SQL
    Server};Database=booknet;Server=ZHANGSHIHUA\TEST;UID=sa;PWD=;"
conn.Open strConn
strSQL = "DELETE FROM Books"
strSQL = strSQL & " WHERE BookID =" & Trim(strBookID)
conn.Execute strSQL,,adCmdText + adExecuteNoRecords
conn.Close
Response.Redirect("index.asp")
%>
```

(2) 运行 index.asp 文件，单击某记录后的“删除”超链接以进入删除记录的操作页面。



图 10-8　删除记录

10.2.7　Supports 方法

Supports 方法用来判断 Recordset 对象是否支持特定的功能，其语法结构如下：

```
Recordset.Supports(CursorOptions)
```

参数 CursorOptions 的取值与相关说明如表 10-6 所示。

表 10-6　Supports 方法参数 CursorOptions 的取值

常 量 名	值	说 明
adAddNew	16778240	确定是否支持 AddNew 方法
adApproxPosition	16384	确定是否可以读写 AbsolutePage 和 AbsolutePosition 属性
adBookMark	8192	确定是否支持 BookMark 属性
adDelete	16779264	确定是否支持 Delete 方法
adHoldRecords	256	确定是否可以提交所有待修改并释放所有当前保留记录，而取回更多的记录或改变下一次取回的记录
adMovePrevious	512	确定是否支持 Move、MoveFirst、MovePrevious 方法
adRecync	131072	确定是否支持 Recync 方法
adUpdate	16809984	确定是否支持 Update 方法
adUpdateBatch	65536	确定是否支持 UpdateBatch 方法

例如，以下脚本用于判断记录集是否支持更新、删除或新增记录：

```
<%If rst.Supports(adUpdate+adDelete+adAddNew) Then%>
```

如果记录集支持的功能与参数 CursorOption 指定的一致，返回 True，否则返回 False。虽然 Supports 方法可以对一个指定的功能返回 True，但数据提供者不能保证在所有环境下这些功能都是可用的。

10.2.8　Status 属性

Status 属性是当前记录集的状态标志，这些标志在记录被更改、删除、插入或改变位置时会受到影响。Status 属性的取值及其相关说明如表 10-7 所示。

表 10-7　Status 属性的取值

常 量 名	值	说 明
adRecOK	0	记录更新成功
adRecNew	1	当前记录是新的，并且还没有更新到数据库
adRecModified	2	当前记录被修改，并且还没有更新到数据库
adRecDeleted	4	当前记录被删除
adRecUnmodified	8	当前记录未被修改
adRecInvalid	16	书签是无效的，记录未被更新
adRecMutipleChanged	64	更改会影响到多条记录，记录未被更新

(续表)

常　量　名	值	说　　明
adRecPendingChanged	128	涉及未确定的插入，记录未被更新
adRecCanceled	256	操作被取消，记录未被更新
adRecCantRelease	1024	记录被锁定，无法进行更新
adRecConcurrencyViolation	2048	记录处于开放式锁定中，无法进行更新
adRecIntegrityViolation	4096	用户违反完整性规则，无法进行更新
adRecMaxChangeExceeded	8192	存在太多的不确定更改，无法进行更新
adRecObjectOpen	16384	与一个打开的存储过程冲突，无法进行更新
adRecOutOfMemory	32768	内存不够，无法进行更新
adRecPermissionDenied	65536	用户没有足够权限，无法进行更新
adRecSchemaViolation	131072	底层数据库会被破坏，记录没有更新
adRecDBDeleted	262144	记录已从数据源中删除

10.2.9　分页显示

Recordset 对象提供了 PageSize、PageCount 和 AbsolutePage 等属性，用来对记录集实现分页显示功能。其中，PageCount 属性确定记录集中包含多少页的记录，PageSize 属性确定每页显示的记录数，AbsolutePage 属性确定当前记录位于哪一页上。

【练习 10-6】按照记录的先后顺序，分页显示数据库中表的内容，程序运行效果如图 10-9 所示。

(1) 确定 adovbs.inc 文件在应用程序的当前目录中。

(2) 建立分页显示和处理表中内容的“分页显示.asp”文件。

```
<%@ Language=VBScript %>
<% Response.Buffer = True %>
<!--#include file="adovbs.inc"-->
<%
Response.Expires = 0
dim strConn,strSQL
Dim intCur,intTotal,I
Dim conn,rs
Const intPageSize = 5
if Request.ServerVariables("CONTENT_LENGTH") = 0 then
    intCur = 1
else
    intCur = CInt(Request.Form("CurPage"))
    Select Case Request.Form("Page")
        Case "首页"
```

```
            intCur = 1
        Case "上一页"
            intCur = intCur - 1
        Case "下一页"
            intCur = intCur + 1
        Case "尾页"
            intCur = CInt(Request.Form("LastPage"))
    End select
end if
Set conn=Server.CreateObject("ADODB.Connection")
strConn="Driver={SQL
    Server};Database=booknet;Server=ZHANGSHIHUA\TEST;UID=sa;PWD=;"
conn.Open strConn
strSQL = "SELECT * FROM Books"
Set rs = Server.CreateObject("ADODB.Recordset")
rs.CursorLocation = adUseClient
rs.CursorType = adOpenStatic
rs.CacheSize = intPageSize
rs.Open strSQL,conn
rs.PageSize = intPageSize
If Not rs.EOF then
    rs.AbsolutePage = intCur
end if
intTotal = rs.PageCount
%>
<HTML><BODY>
<Center><H4>分页显示数据表中的内容</H4>
<Hr>
<B>共有图书<%=rs.RecordCount %>本  共<%=intTotal%>页  当前是第<%=intCur%>页</B>
<p>
<FORM NAME="" METHOD=POST
      ACTION="<%=Request.ServerVariables("SCRIPT_NAME")%>">
<INPUT TYPE="hidden" NAME="CurPage" Value="<%=intCur%>">
<INPUT TYPE="hidden" NAME="LastPage" Value="<%=intTotal%>">
<INPUT TYPE="Submit" NAME="Page" Value="首页">
<% if intCur > 1 then %>
    <INPUT TYPE="Submit" NAME="Page" Value="上一页">
<% end if %>
<% if intCur <> intTotal then %>
    <INPUT TYPE="Submit" NAME="Page" Value="下一页">
<% end if %>
<INPUT TYPE="Submit" NAME="Page" Value="尾页">
</FORM>
```

```
</p>
<%
With Response
  if rs.EOF then
      .Write "there are no record"
  else
      .Write "<TABLE BORDER=1 CELLSPACE=0 CELLPADDING=5>" &_
          "<TR HEIGHT=12><TH WIDTH=40> 序号 </TH>" &_
          "<TH WIDTH=300> 书名 </TH>" &_
          "<TH WIDTH=200> 作者 </TH>" &_
          "<TH WIDTH=40> 价格 </TH></TR>"
  end if
  I = 0
  do While not rs.EOF and I < rs.PageSize
      .Write "<TR HEIGHT=12><TD WIDTH=40>" & rs.AbsolutePosition & "</TD>" &_
        "<TD WIDTH=300>" & rs("BookName") & "</TD>" &_
        "<TD WIDTH=200>" & rs("Writer") & "</TD>" &_
        "<TD WIDTH=40>" & rs("Price") & "</TD></TR>"
        rs.MoveNext
      I = I + 1
  loop
  .Write "</TABLE>"
End With

rs.Close
conn.Close
%>
<HR>
</Center>
</BODY></HTML>
```

(3) 网页运行后效果如图 10-9 所示。



图 10-9　分页显示

10.3　认识 Fields 集合与 Field 对象

每个 Recordset 对象都包含一个 Fields 集合，该集合用来处理记录集中的字段。记录集中返回的每个字段在 Fields 集合中都有一个对应的 Field 对象。通过 Field 对象，可访问字段名、字段类型、字段值等信息。

10.3.1　Fields 集合简介

Fields 集合只有一个 Count 属性和一个 Item 方法。Count 属性返回记录集中字段(Fields 对象)的个数，Item 方法用于建立某一个 Field 对象。

1. Fields 集合的属性

Fields 集合只有一个 Count 属性，该属性返回记录集中字段(Fields 对象)的个数，其使用方法为：

```
Var=Recordset 对象.Fields.Count
```

Var 的值为字段的个数。如下例将使用脚本列出当前记录中每个字段的值。

```
For I=0 To rst.FieldsCount-1
Response.Write rst(I)&"<Br>"
Next
```

2. Fields 集合的方法

Fields 集合只有一个 Item 方法，该方法用于建立某一个 Field 对象，其使用方法为：

```
Set Field 对象=Recordset 对象.Fields.Item(字段名或字段索引值)
```

其中，字段索引值是根据记录集中的先后顺序排序，从 0~FieldsCount－1。如下例将通过名字或序号返回相关的字段的值：

```
rs.Fields.Item (2)
rs.Fields.Item ("Writer")
```

Item 方法是 Fields 集合的默认方法，而 Fields 集合是 Recordset 对象的默认集合。因此，Item 和 Fields 可省略，如上例可简化为：

```
rs (2)
rs ("Writer")
```

10.3.2　Field 对象简介

下面将简单介绍 Field 对象的属性和方法。

1. Field 对象的属性

Field 对象的属性及其说明如表 10-8 所示。

表 10-8　Field 对象的属性

属　　性	说　　明
ActualSize	返回字段的实际长度
Attributes	返回字段的特征，可取下列值之一或组合。 ● 2 或 adFldMayDefer：只有在明确指明要取这个字段时，该字段的值才从数据源返回 ● 4 或 adFldUpdatable：字段可更新 ● 8 或 adFldUnknownUpdatable：无法确定字段是否可更新 ● 16 或 adFldFixed：字段包含固定长度的数据 ● 32 或 adFldIsNullable：字段可以接收空值，可用于检测非空字段 ● 64 或 adFldMayBeNull：可以从该字段读取空值 ● 128 或 adFldLong：字段类型为长整型 ● 256 或 adRowID：字段是一个行标识符 ● 512 或 adFldRowVersion：字段包含跟踪更新日期和时间的标记 ● 4096 或 adFldCacheDeferred：数据源将字段放入高速缓存中 用法：if rs("Writer").Attributes and adFldIsNullable then… 即若字段允许接收空值，则执行此段代码
DefinedSize	返回字段的定义大小，如 rs("Writer").DefinedSize
Name	返回数据库中字段的名字
NumerucScale	说明字段的小数部分需要多少个数字位
Precision	说明字段的值需要多少个数字位
OriginalValue	返回字段修改前的值
UnderlyingValue	从数据库返回字段的当前值
Type	字段的数据类型
Value	返回字段的值

2. Field 对象的方法

Field 对象有 AppendChunk 和 GetChunk 两个方法。AppendChunk 方法用于将大块文本和二进制数据写到字段中。当系统内存有限时，可用 AppendChunk 方法将数据分块写入，

而不用一次将数据完全添加进去。该方法的使用方法是：

```
Object.AppendChunk Data
```

其中，Data 是待写入字段中的数据。每次后续的 AppednChunk 调用总是把数据添加到已有数据的后面。如果正在添加数据到一个字段中，同时又去操作当前记录的其他字段值，则 ADO 会认为对这个字段添加数据的操作已完成。此时，再对这个字段调用 AppendChunk 方法，ADO 会认为是一个新的 AppendChunk 调用，并覆盖掉已有的数据。

GetChunk 方法从字段中获取大块文本或二进制数据的部分或全部内容。当系统内存有限时，可用 GetChunk 方法将数据分块读出，而不用一次全部读出来。该方法的使用方法是：

```
Object.GetChunk(Size)
```

其中，Size 是待获取的字节数或字符数。如果 Size 比余下的数据大，GetChunk 方法仅返回剩余的数据；如果字段是空的，则返回 Null。

每次后续的 GetChunk 方法调用总是从前一次调用结束的位置开始。但如果正从一个字段读取数据，然后又去操作当前记录的另一个字段的值，则 ADO 会认为这个字段的读取数据的操作已完成。此时，再对这个字段调用 GetChunk 方法，ADO 会认为是一个新的 GetChunk 调用，即重新从开始位置读起。

当 Field 对象的 Attribute 属性的 adFldLong 被置为 True 时，可在该字段上使用 GetChunk 和 AppendChunk 方法。

10.4 认识 Errors 集合与 Error 对象

涉及 ADO 对象的操作可能产生一个或多个错误，这些错误都和数据提供者有关。当错误发生时，就会有一个或多个 Error 对象被放置到 Connection 对象的 Errors 集合中。当另外一个 ADO 操作产生错误时，将清除 Errors 集合，并把新的 Error 对象集放到 Errors 集合内。

Errors 集合有一个 Count 属性，该属性用来指出 Errors 集合目前所包含的 Error 对象的个数。Errors 集合有一个 Item 方法，可以调用 Item 方法从 Errors 集合中获得某个具体的 Error 对象。Item 方法是 Errors 集合的默认方法，调用时可省略不写。此外，还可调用 Clear 方法从 Errors 集合中清除所有的 Error 对象。

Error 对象的属性及其相关说明如表 10-9 所示。

表 10-9 Error 对象的属性

属　性	说　明
Description	关于该项错误的描述文字
HelpContext	这是一个长整数，对应于产生错误对象的相关帮助主题
HelpFile	对应于产生错误对象的相关帮助文件
NativeError	数据提供者所提供的针对错误的错误代码

(续表)

属　性	说　明
Number	ADO 所产生的错误代码
Source	错误所产生的对象名称
SQLState	该属性是一个长度为 5 个字节的字符串，包含按 SQL 标准所定义的错误

10.5　认识 Properties 集合与 Property 对象

Connection、Recordset、Field、Command 对象都有 Properties 集合。Properties 集合用来保存与这些对象有关的各个 Property 对象。Property 对象表示各个选项设置或其他没有被对象的固有属性处理的 ADO 对象特征。

ADO 对象一般包含两种类型的属性：固有属性和动态属性。固有属性不是作为 Property 对象出现在 Properties 集合中。当创建新的 ADO 对象后，这些固有属性可立即使用。例如，可以用 Recordset 对象的 EOF 和 BOF 属性来判断当前记录是否已到达边界。

动态属性是由后端数据提供者定义的，这些属性被放到 Properties 集合中。每个特定的 ADO 对象都有一个 Properties 集合。

Properties 集合有一个 Count 属性，用来指出 Properties 集合上有多少个 Property 对象。可以用 Item 方法从 Properties 集合中获得某个 Property 对象。Item 方法是 Properties 集合的默认方法，调用时可省略不写。如果要从数据提供者中取得 Properties 集合和扩展的属性信息，可以调用 Refresh 方法。

Property 对象的属性及其相关说明如表 10-10 所示。

表 10-10　Property 对象的属性

属　性	说　明
Attributes	指定何时和如何设定 Property 对象的值
Name Property	Property 对象的名称
Type Property	指定当前 Property 对象值的数据类型
Value Property	设置 Property 对象的值

10.6　习　　题

10.6.1　填空题

(1) Recordset 中的记录指针具有游标类型，它的默认值为____________。

(2) ____________是针对数据库操作中并发事件的发生而提出的系统安全控制方式。

(3) Recordset 对象提供 4 个重要的数据浏览方法：MoveFirst、____________、____________和____________，分别将当前的指针移到记录集的首记录、前一个记录、后一个记录和末记录，

(4) Fields 集合的____________属性返回记录集中字段(Fields 对象)的个数，____________方法用于建立某一个 Field 对象。

10.6.2　选择题

(1) Recordset 对象在使用前需要使用____________对象建立数据库的连接。

A. Application　　　　B. Connection

C. Sever_OnStart　　　　D. Session

(2) 在 Recordset 对象创建完成之后，即可打开____________对象的内容。

A. Application　　　　B. Connection

C. Recordset　　　　D. Session

10.6.3　问答题

(1) 在使用 INSERT 语句添加数据时需要注意哪些方面？

(2) Recordset 对象使用 Connection 对象与数据库建立连接的操作步骤是什么？

10.6.4　操作题

(1) 参考本章【练习 10-1】的操作创建一个 ASP 查询页面。

(2) 参考本章【练习 10-2】的操作创建一个能够在数据库中添加记录的 ASP 页面。

第11章　XML 与 ASP

在支持 WWW(万维网)的技术中，可扩展标记 XML 被认为是最重要的新技术之一。虽然它已经对数据的存储和处理产生了深远的影响，但它的优势在将来的某个时候才会全面体现。XML 实际上是一系列相关技术的集合，这些技术源于万维网联盟(Word Wide Web Consortium,W3C)对 WWW 技术的发展规划。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 了解 XML 与 HTML 的区别
- 了解 XML 文档的结构
- 掌握创建 XML 文档的方法
- 掌握使用 ASP 操作 XML 的方法

11.1　认识 XML

XML(Extensible Markup Language)用来创建在 Web 页面和应用程序之间定义和传输的数据。它是一种描述数据和数据结构的语言，可以保存在任何可以存储文本的文档中。XML 具有自描述性、内容与显示相分离、可扩展性、独立于平台等特点，提供了一套跨平台、跨网络、跨程序语言的数据描述方式，已经逐渐成为科学与业务应用中数据表示及应用的数据交换标准格式。

11.1.1　标记语言的发展

标记语言包括文本、图片和其他多媒体链接，可以链接到当前文档的其他部分，或者其他文档和其他对象等。标记是指编辑器在待修改的文档上做的标签。标记由元素、属性、实体和符号组成。元素定义了文档的块；属性规范增强这些块；实体是一个或多个字符，通常在键盘上找不到；符号是记号或者字符集，它们构成了一个字母表。文档中没有标记的每个元素的内容，通常是文本(包括图片)。

标记语言在发展过程中经过了以下几个阶段：

- SGML：1986 年标准通用语言(SGML)成为 ISO 的标准。因为 SGML 是标准语言，世界上各类商业组织用它来发布各类文档及创建自己的标记语言。例如，HTML 就是用 SGML 创建的最流行的标记语言。SGML 文档包括了文本和多媒体元素，也包括了各类标题、段落和一些用于格式化的元素。每一个 SGML 文档都有一个文

档类型声明，包括文档类型定义(DTD)。DTD 定义了文档内容的规则。此外，DTD 为一个或多个文档指定一种标记语言。

- HTML：HTML 语言被定义用来处理 Web 上的超媒体功能。在一个简单的文本文档中，HTML 开发者插入一个元素，将站点链接到另一个站点，或浏览器、搜索引擎、计算机、网络和描述文档。
- XML：可扩展性标记语言(XML)是 SGML 的一个子集。XML 包含了很多 SGML 特性，但是比 SGML 简单得多。像 SGML 一样，用户可以使用 XML 来开发一种标记语言，它的元素和属性是为专门行业和产业而定义的。创建这种语言之后，用户就可以像使用 HTML 一样，使用 XML 来标记并结构化文档。

目前，XML 和 HTML 都支持统一字符编码协会(Unicode Consortium)制定的通用字符集，不仅包括特殊字符、标点和数学符号，还包括非英语语言的字母和字母表，这使得 XML 成为了国际标准。

11.1.2　XML 的特点

XML 实际上是 Web 上表示结构化信息的一种标准文本格式，没有复杂的语法和包罗万象的数据定义。XML 的特点可以归纳为 3 点：先进特性，灵活性和自描述性。

- XML 的先进性：XML 继承了 SGML 的许多特性，首先是可扩展性。XML 允许使用者创建和使用他们自己的标记而不是 HTML 的有限词汇表。这一点至关重要，企业可以用 XML 为电子商务和供应链集成等应用定义自己的标记语言，甚至与特定行业一起来定义该领域的特殊标记语言，作为该领域信息共享与数据交换的基础。
- XML 的灵活性：HTML 很难进一步发展，就是因为它是格式、超文本和图形用户界面语义的混合，要同时发展这些混合在一起的功能是很困难的。而 XML 提供了一种结构化的数据表示方式，使得用户界面分离于结构化数据。所以，Web 用户所追求的许多先进功能在 XML 环境下更容易实现。
- XML 的自描述性：XML 文档通常包含一个文档类型声明，因而 XML 文档是自描述的。不仅用户能读懂 XML 文档，计算机也能处理。XML 表示数据的方式真正做到了独立于应用系统，并且数据能够重用。XML 文档被看作是文档的数据库化和数据的文档化。

另外，XML 还具有简明性。它只有 SGML 约 20%的复杂性，但却具有 SGML 约 80%的功能。XML 比完整的 SGML 简单得多，易学，易用并且易实现。

11.1.3　XML 与 HTML 的区别

虽然 XML 和 HTML 都使用标记，但是他们是不同的。其中最主要的区别是 XML 专门用来描述文本的结构，而不是用于描述如何显示文本。XML 不像 HTML 一样具备固定

的标记，例如下面所示的代码，如果被看作是 HTML 文档，在浏览器中将显示为如图 11-1 所示的效果。

```
<body>
简单的代码
<h1>一级标题</h1>
普通文本
<b>加粗文本</b>
下面是其他文本
</body>
```

但是，以上代码如果被作为 XML 文档，其中的标记就不具备任何含义其内容仅是说明，在浏览器中打开可以查看到如图 11-2 所示的结果。

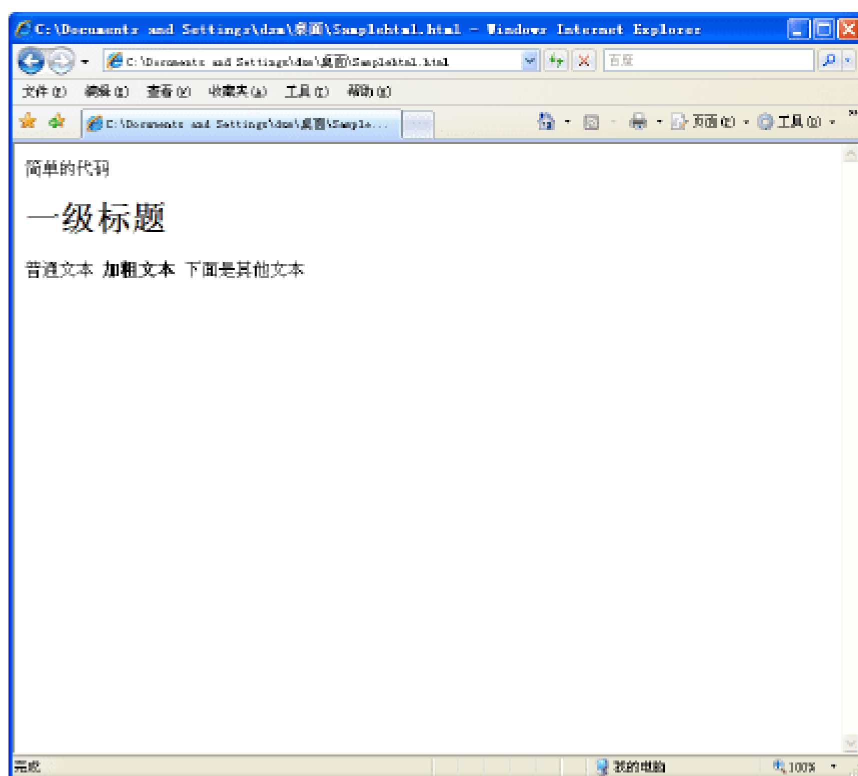


图 11-1　浏览器中显示的 HTML 文档

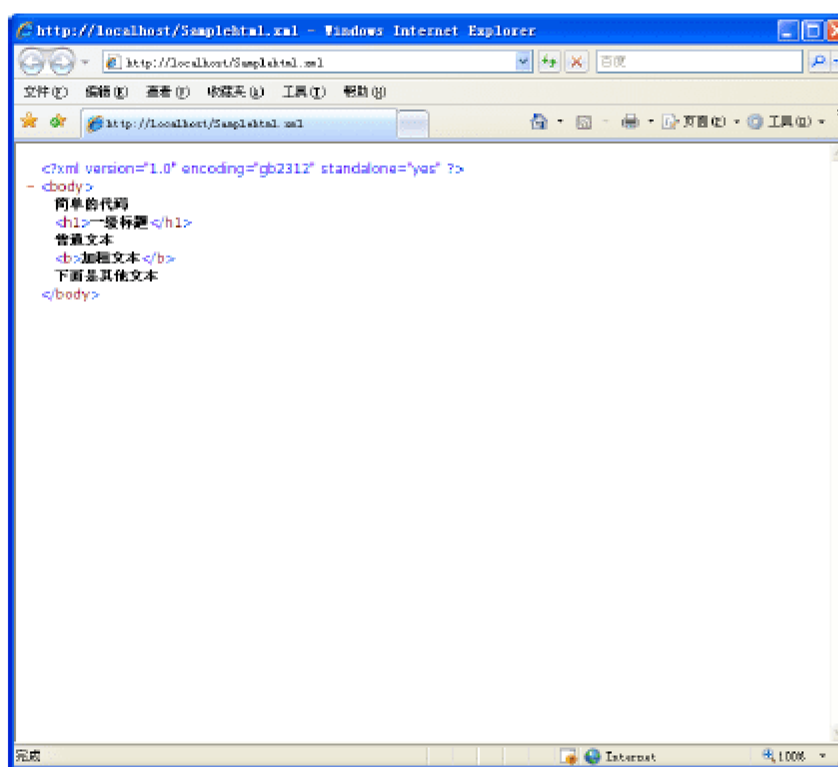


图 11-2　浏览器中显示的 XML 文档

从图 11-2 中可以看出，浏览器在解析 XML 文档时，并没有对 XML 文档进行任何的处理，仅仅是将其显示出来而已。

【练习 11-1】创建简单的 XML 文档，并在编写代码时使用几种不同的标记。通过在浏览器中查看 XML 文档效果，了解 XML 中如何使用数据。

(1) 选择“开始”|“所有程序”|“附件”|“记事本”命令，打开记事本工具。

(2) 输入以下代码(如图 11-3 所示):

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<Orders>
<Order>
<ID>20060911a</ID>
<Name>标准实例</Name>
<Price>29.8</Price>
<Quantity>10</Quantity>
</Order>
<Order>
```

```
<ID>20060911b</ID>
<Name>应用实例</Name>
<Price>29.8</Price>
<Quantity>25</Quantity>
</Order>
</orders>
```

(3) 输入的代码保存为 XML 文档后在浏览器中打开，效果如图 11-4 所示。

(4) 在如图 11-3 所示的程序代码中使用了几种不同的标记。对于初学者而言，可能会认为这些标记各有其含义。它们都有一个有意义的名称，该例定义了订单的一个列表，单个订单以及一些与订单有关的值。在浏览网页时，如果上面的代码是 HTML 可以把它格式化成一个表来显示，但因为这里是 XML，XML 中的标记不代表任何意义。



图 11-3　Order.xml 的代码

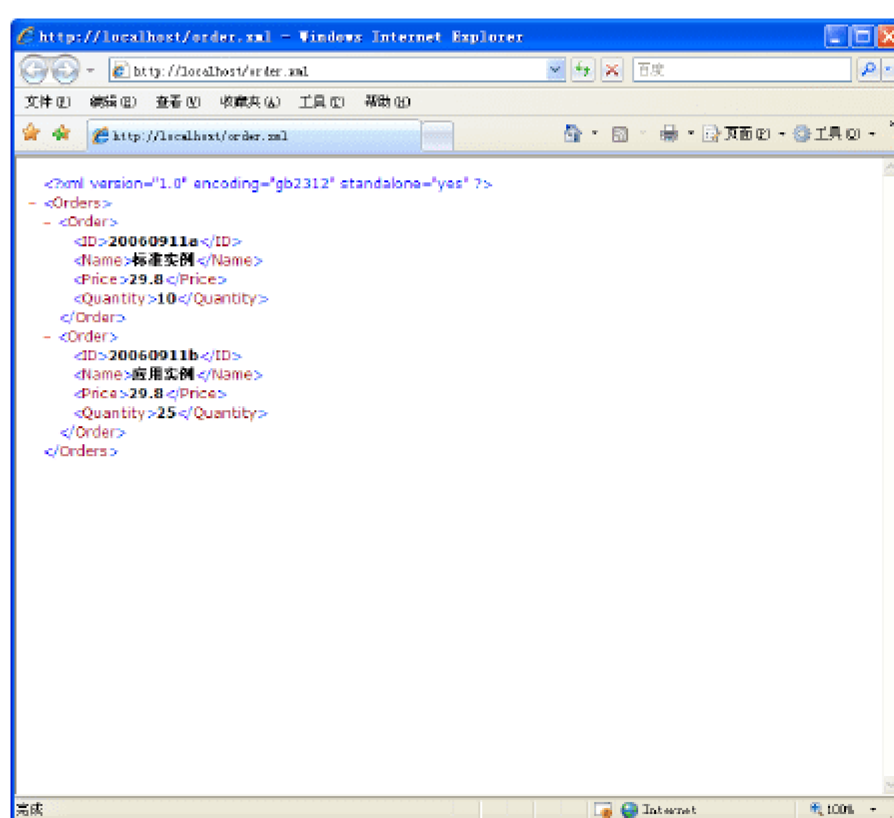


图 11-4　XML 的效果

(5) 通过上面的实例代码可以看出，浏览器没有对代码进行任何处理。因为即使代码中的标记对用户来说具备意义，但它们对于 XML 却没有意义。下面把图 11-3 所示的代码修改为如下所示(如图 11-5 所示)，并在浏览器中显示，如图 11-6 所示。

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<图书订单>
<订单>
<订单编号>20060911a</订单编号>
<图书名称>标准实例</图书名称>
<价格>29.8</价格>
<数量>10</数量>
</订单>
<订单>
<订单编号>20060911b</订单编号>
<图书名称>应用实例</图书名称>
<价格>29.8</价格>
<数量>25</数量>
```

```
</订单>
</图书订单>
```



图 11-5　修改后的代码

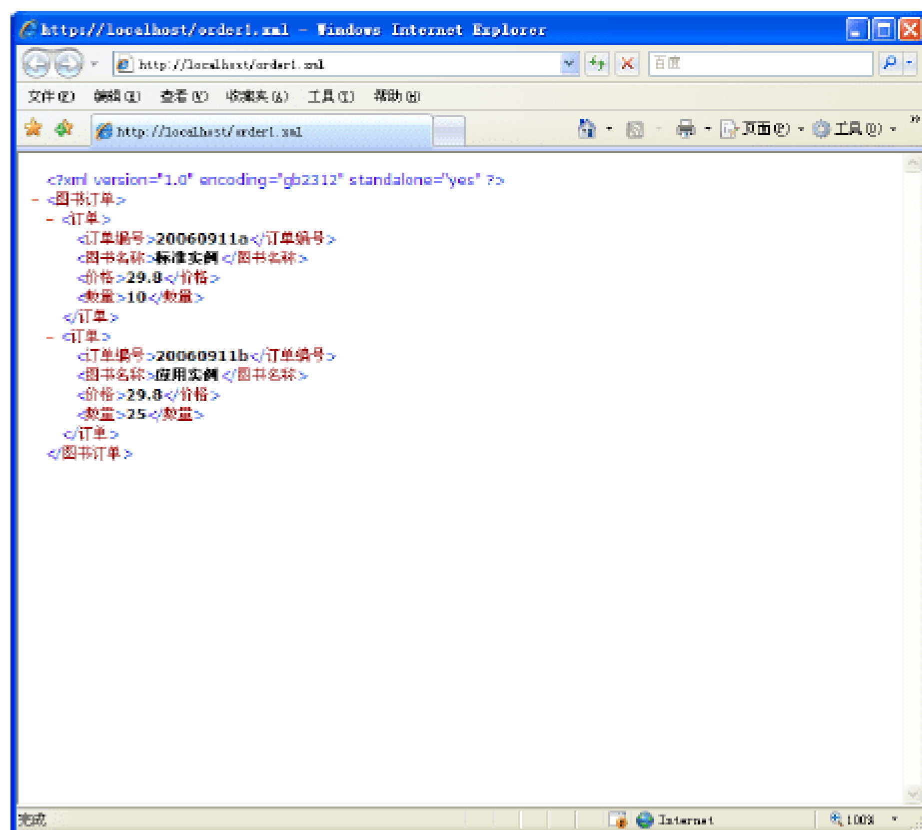


图 11-6　修改代码后的网页效果

注意：

通过上面实例的效果图 11-4 和图 11-6 可以看出，XML 由一系列能描述文档各个部分的标记组成。在图 11-3 所示程序代码中使用 XML 来描述数据，使用了代表数据字段名的标记名。这就是 XML 能作为一种数据交换格式的真正含义。

11.2　XML 文档的结构

早期版本的 HTML 文档的顶端和底部各需要一个<HTML>元素，其中定义了文档的两个主要部分的嵌套元素<HEAD>和<BODY>。XML 文档也遵循类似 HTML 文档一样的两部分格式，即声明在文档的顶端，实体是文档的主体

11.2.1　XML 声明

每一个 XML 文档都应该以 XML 声明开始，声明的作用是指定所使用的 XML 版本，其完整的语法如下所示：

```
<?xml version = "1.0"?>
<!DOCTYPE sampledoc [
<!ELEMENT greeting(#PCDATA)>
]>
```

上面例子是一个内部文档类型定义，可作为一个组织完好的文档声明。

DTD 设置所有的元素、属性和其他的文档组件的规则。例如，下面的语句引用了一个外部 DTD：

```
<?xml version = "1.0"?>
<!DOCTYPE sampledoc SYSTEM "sample.dtd">
```

在 XML 声明中可以使用 3 个属性，分别是版本，独立和编码。这 3 个属性均为可选项，但应该至少正确包含版本属性，来指明 XML 的版本号。

1. 版本属性

下面的语句为只包含版本属性的 XML 声明：

```
<?xml version = "1.0"?>
```

上述的 XML 声明实际上并不是标记，而是一个处理指令，是将信息传递给处理 XML 文档的应用程序的特殊语句。通常处理指令很容易识别，都是以“<?”开始，以“?>”结束。

2. 编码属性

XML 声明的编码属性指明了 XML 文档使用的语言。虽然英语是使用在 Web 上的主要语言，但并不是唯一的语言。使用 XML 声明编码属性，为 XML 文档指定字符集，就应该在文档中使用以下 XML 声明：

```
<?xml version = "1.0" encoding = "utf-8"?>
```

3. 独立属性

独立属性用来指定正确显示一个文档时是否需要文档类型定义(DTD)。基于 SGML(标准通用标记语言)的语言使用 DTD 定义后(例如 HTML)，可以在文档中使用元素和属性。

与其他基于 SGML 的语言不同，要正确显示 XML 文档并不要 DTD 定义，因为 XML 不包含任何预定义元素。而且与 XML 文档有关的标记是定义在文档本身内的，而不是定义在单独的 DTD 中，所以 XML 是自描述性的不需要 DTD。

但是，一个 XML 文档也可以使用 DTD 定义从而获得更多的特性，尤其是当多重 XML 文档共享相同的元素时，如果 XML 文档需要 DTD，可以为独立属性赋予值“否”。但是，如果确定 XML 文档不需要 DTD，则可以将独立属性赋值为“是”，例如上面创建的 XML 文档就是对任何不需要 DTD 的 XML 文档，它的声明如下：

```
<?xml version = "1.0" encoding = "utf-8" standalone="yes"?>
```

注意：

许多 XML 文档包含一个文档类型(!DOCTYPE)声明，或者包含一个内嵌于文档的文档类型定义，或者指定一个外部的 DTD，文档类型声明由保留字!DOCTYPE 开始，声明了文档的类型。

11.2.2　XML 实例

XML 文档实例包括 XML 文档余下的部分，就是除了声明以外的所有部分(注意，有些 XML 开发者实例包含了整个文档)。实例包含了元素和属性的定义以及实例和内容。

1. 元素和标记

元素和标记不是同一个概念。一个元素由一个开始标记开头，由一个终止标记结尾，在起始标记和终止标记之间就是元素名称、可选属性和其他内容，如下所示：

```
<font size ="13px">数据资源 安全备份</font>
```

注意：

在 HTML 中，有些元素允许忽略终止标记。但是在 XML(以及 XHTML)中，终止标记是必需的。因此，如果现在使用 HTML 文档并且将来打算转换为 XML 文档的话，为了保持与 XML 的兼容，在 HTML 文档中使用终止标记是一个很好的习惯。

2. 空元素

在 XML 中，也支持空元素标记，即没有内容的元素。空元素指的是一个文档输出时加入的对象，如图片和分割线。空元素支持两类标记语法：一类是使用与非空元素相同的语法，另一类则是空元素标记特有的语法，例如：

```
<image />  或者<br />
```

注意：

空元素的第 2 种语法中，典型的终止标记连在一起的斜杠出现在标记的最后，前面是一个空格。空元素语法实现上包含一个起始标记和一个终止标记。

空元素通常都有属性。例如在一个文档处理之后要加入一个图片，那么就必须有一个参照点，像图片的来源，如下所示：

```
<image src="school.gif" />  或者<image src="school.gif"></image>
```

另外，用户还可以使用一个没有实际内容的起始标记和终止标记，如下所示：

```
<BR> <BR/>
```

3. 属性

属性允许用户定义元素的性质和特性。例如，可以从一个特定的值开始一个数字列表，或者为图片添加一个标记符，这样就可以在文档中很容易地找到它——用于链接或指定一个格式或增强区域的开头(或结尾)。带属性的元素形式如下：

```
<element option1="value1"    option2="value2"></element>
```

注意，这里的小于号和大于号之间的起始标记中包括了所有的属性，如下所示：

```
<image src="school.gif"   width="200"   height="200" border="1" align="center"> </image>
```

11.2.3　创建和解析 XML 文档

前面所介绍的内容都是与 XML 文档结构相关的，并没有创建或者使用 XML 文档。下面将介绍如何创建并在浏览器中解析 XML 文档的方法。

XML 文档，既可以在例如记事本或写字板之类的文本编辑工具中创建，也可以在任何能够创建简单文本文件的字处理软件或网页编辑软件中建立。如果用户使用文本编辑器创建 XML 文档，需要在解析器中打开 XML 文档，才能查看网页的最终效果，如果使用一些专用的 XML 编辑器(例如 Microsoft XML Notepad)创建 XML 文档，即可即时查看网页效果。

1. 创建 XML 文档

下面将通过实例介绍在记事本工具中创建 XML 文档的方法。

【练习 11-2】通过创建一个简单 XML 文档，了解创建 XML 文档的方法。

(1) 选择“开始”|“所有程序”|“附件”|“记事本”命令，打开记事本工具。

(2) 在记事本工具中输入一行 XML 文档的声明语句，并在声明语句中指定 XML 文档的版本、编码属性和独立属性，如下所示：

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
```

(3) 在记事本文档中输入 XML 文档的实例，如下所示。

```
<Orders>
<Order>
<ID>20060911a</ID>
<Name>标准实例</Name>
<Price>29.8</Price>
<Quantity>10</Quantity>
<Date>2007-09-03</Date>
<User>ZHT</User>
<Address>Nanjing</Address>
</Order>
</Orders>
```

(4) 完成代码的输入工作后，选择“文件”|“保存”命令，将代码以文件名 Order2.xml 保存即可完成 XML 文档的创建工作。

2. 解析 XML 文档

在浏览器中打开一个没有结束标记</html>的 HTML 文档时，浏览器会简单地忽略该错误并显示网页的效果。而 XML 文档则必须符合严格的要求才能在浏览器中显示(其中最重要的规则是所有的标记必须全部关闭)。当文档符合 XML 文档的格式正确时，其格式就是正确的。

XML 文档格式是否正确，可以使用解析器程序进行检查。解析器程序有非检验解析器和检验解析器两种类型。所谓非检验解析器只简单地检验 XML 文档格式是否正确。而检验解析器不但检验 XML 文档格式是否正确，还检验 XML 文档是否遵循相关的 DTD。

【练习 11-3】以【练习 11-2】创建的 XML 文档为基础，通过 Internet Explorer 浏览器解析创建的 XML 文档。

(1) Internet Explorer、Netscape 以及其他浏览器具备作为非检验解析器的能力。如果用户使用 Internet Explorer 浏览器打开【练习 11-2】所示的 XML 文档，如果非检验解析器发现文档格式正确，就显示文档的 XML 标记和数据，如图 11-7 所示。

(2) 如果非检验解析器发现 XML 文档不正确(例如将【练习 11-2】创建文档中的代码内的<./Orders>标记去掉再运行)，浏览器在运行文档时将解析出错误，并在页面中指出错误，效果如图 11-8 所示。

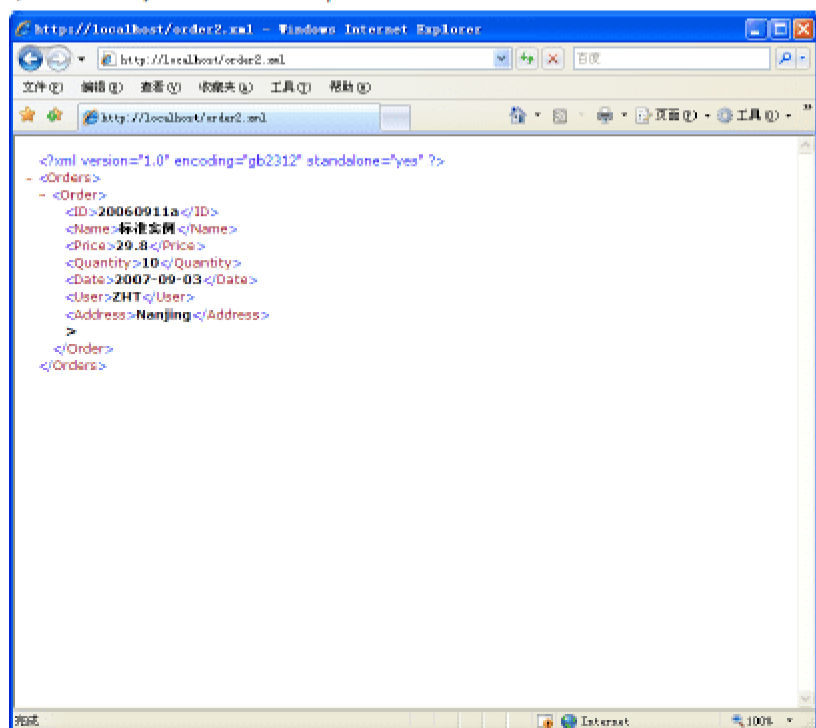


图 11-7　IE 浏览器显示格式正确的 XML 文档

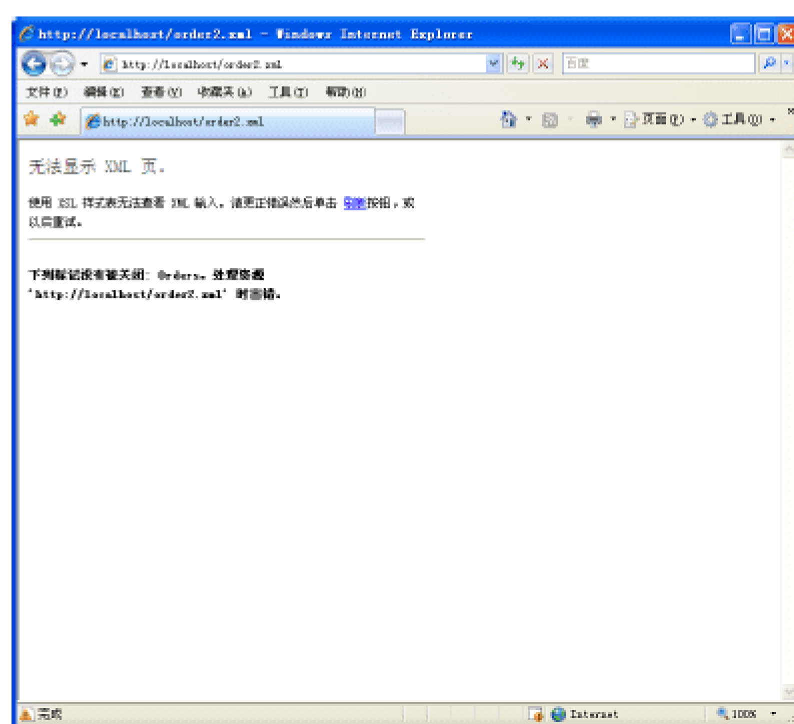


图 11-8　IE 浏览器解析错误文档

11.2.4　编写正确的 XML 文档

创建 XML 文档的目的是为了更好地组织数据。编写正确的 XML 文档可以让用户代理和 Web 应用程序更容易读取文档中的数据。

1. 文档必须具有根元素

根元素在一个页面中包含其他元素。HTML 文档中的根元素一般是<HTML>和</HTML>。而 XML 文档则需要用户自己定义根元素(例如在本章【练习 11-2】中创建的 XML 文档中，根元素为<Orders>)。如果在文档中没有包含根元素，那么 XML 文档的格式就不正确。例

如，下面所示的 XML 文档就不符合文件必须具备根元素的规则：

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<ID>20060911a</ID>
<Name>标准实例</Name>
<Price>29.8</Price>
<Quantity>10</Quantity>
<Date>2007-09-03</Date>
<User>ZHT</User>
<Address>Nanjing</Address>
```

注意：
如果 XML 文档没有根元素，在 IE 浏览器中解析时就会出现如图 11-8 所示的错误。

2. 区分文档大小写

与 HTML 标记不同，XML 标记区分大小写，HTML 标记则不区分大小写。例如，在 HTML 文档中下弧线的标记是大写还是小写对于网页最终的效果没有区别。在浏览器中，下面 4 种 HTML 语句都能正确地显示。但是，在 XML 文档中，如果有一个标记写为小写，那么必须在结束标记中全部使用小写，下面代码中混合大小写两种标记的后两条语句将会被认为是完全不同的标记，在文档中造成不正确的格式。

```
<u>HTML 下划线标记-小写</u>
<U>HTML 下划线标记-大写</U>
<u>HTML 下划线标记-小写</U>
<U>HTML 下划线标记-大写</u>
```

3. 必须具备结束标记

在 HTML 文档中，如果代码没有正确组织，大多数浏览器通常会忽略。例如，段落标记<p>用来标识作为单独一段的一块文本，这块文本包含在<p>和</p>标记中，如下所示：

```
<p>HTML 文档标记实例</p>
```

但是，许多时候用户不必遵循这一原则，只需要简单地在文本块末尾使用一个<p>标记来创建一个段落即可，如下所示：

```
HTML 文档标记实例<p>
```

以上方法在 HTML 文档中可行的原因之一是，浏览器通常将 HTML 文档作为包含格式化标记的文本对待。但是 XML 是用来组织数据，而不是用来显示数据的。因此，XML 文档由包含文本的标记组成，而不是由包含标记的文本组成。

4. 正确嵌套 XML 元素

在 HTML 文档中，如何嵌套标记没有区别。例如下面的 HTML 语句是正确的：

```
<font size="18"><div align="center">
<a href="index.asp">首页</font></a>
</div>
```

在上面的代码中，第 3 个标记<a>被认为是最内层的元素。在 XML 中，每个最内层元素必须在其他元素结束之前结束。但是在前面的代码语句中，<font>标记在<a>标记结束前就结束了。这些在 HTML 中并没有区别，但是在 XML 中必须写成如下所示的形式：

```
<font size="18"><div align="center"><a href="index.asp">首页</a></div></font>
```

例如，在下面所示的 XML 文档中包含了订单数据。代码格式不正确，因为<Address>元素未正确地嵌套在<User>元素中。

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<Order>
<ID>20060911a</ID>
<Name>标准实例</Name>
<Price>29.8</Price>
<Quantity>10</Quantity>
<Date>2007-09-03</Date>
<User>ZHT<Address> Nanjing</User></Address>
</Order>
```

如果要使上面所示的代码的格式正确，<Address>元素必须在<User>元素之前结束，如下所示：

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<Order>
<ID>20060911a</ID>
<Name>标准实例</Name>
<Price>29.8</Price>
<Quantity>10</Quantity>
<Date>2007-09-03</Date>
<User>ZHT<Address> Nanjing</Address></User>
</Order>
```

5. 属性值必须加引号

在 HTML 文档中，赋给属性的值可以包含在引号内，也可以不必使用引号直接赋给属性。例如，常见的 HTML 中的表格标记<table>的 width 属性。将要创建表格的宽度赋值给 width 属性。下面的两条代码显示了两个<table>标记，虽然第一个标记的 width 属性值使用了引号，而第二个标记没有使用，但是它们的运行效果是相同的。

代码 1：

```
<table width="850" height="57" border="0" cellpadding="0" cellpadding="0"
    background="image/00".jpg>
```

代码 2：

```
<table width=850 height=57 border=0 cellpadding=0 cellpadding=0 background=image/00.jpg>
```

但是，在 HTML 中必须为赋值给属性的值添加引号。例如，如果用户要给【练习 11-2】创建 XML 订单文档中的<User>元素添加一个属性值 Type，这个属性保存了生成订单用户的类别，如图 11-9 所示。

打开 XML 文件后，在<User>元素中添加属性 Type，如下所示：

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<Orders>
<Order>
<ID>20060911a</ID>
<Name>标准实例</Name>
<Price>29.8</Price>
<Quantity>10</Quantity>
<Date>2007-09-03</Date>
<User Type="Member">ZHT</User>
<Address>Nanjing</Address>>
</Order>
</Orders>
```

在浏览器中解析上面代码时，如果省略了 Type 属性的引号，就会导致文档的格式不正确，如图 11-10 所示(在解析时会出现找不到左引号错误)。

打开 XML 文件后，在<User>元素中添加属性 Type，但省略 Type 属性的引号，如下所示：

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<Orders>
<Order>
<ID>20060911a</ID>
<Name>标准实例</Name>
<Price>29.8</Price>
<Quantity>10</Quantity>
<Date>2007-09-03</Date>
<User Type=Member>ZHT</User>
<Address>Nanjing</Address>>
</Order>
</Orders>
```

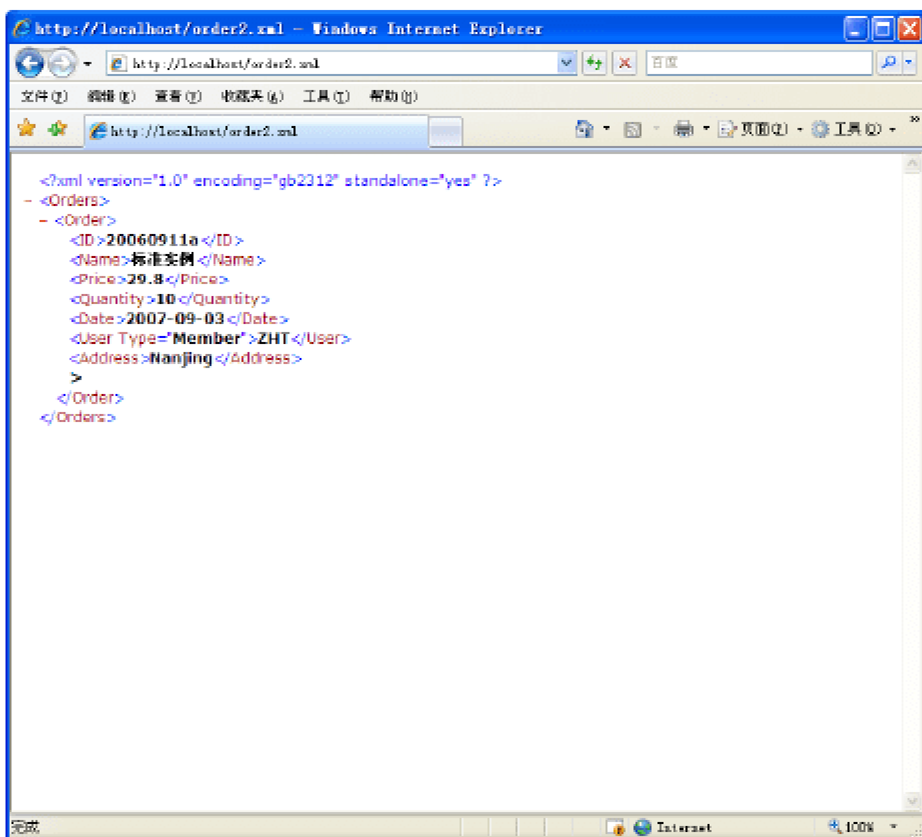


图 11-9　为属性添加引号后的 XML 文档

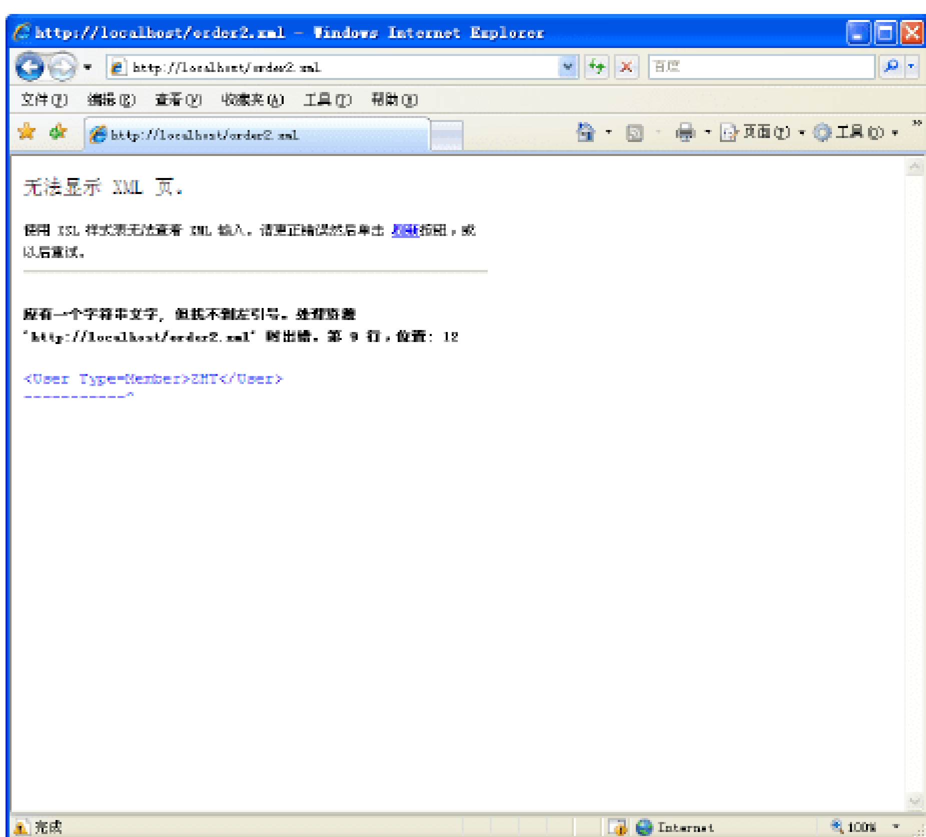


图 11-10　属性值没有引号导致的错误

6. 空元素必须关闭

在 HTML 中，许多元素都没有对应的结束标记，例如<hr>和
等。一般将这些不需要结束标记的元素称为空元素。下面为【练习 11-2】创建的订单 XML 文档创建一个带有属性的<PayOut>空元素，这个属性保存了一个逻辑值，用于表示订单是否支持货到付款。

<PayOut>空元素的 XML 代码如下所示：

```
<PayOut Boolean="True" />
```

完成【练习 11-2】创建的 XML 文档的编辑后(如图 11-11 所示)，XML 文档在浏览器中的显示效果如图 11-12 所示。



图 11-11　编辑 XML 文档

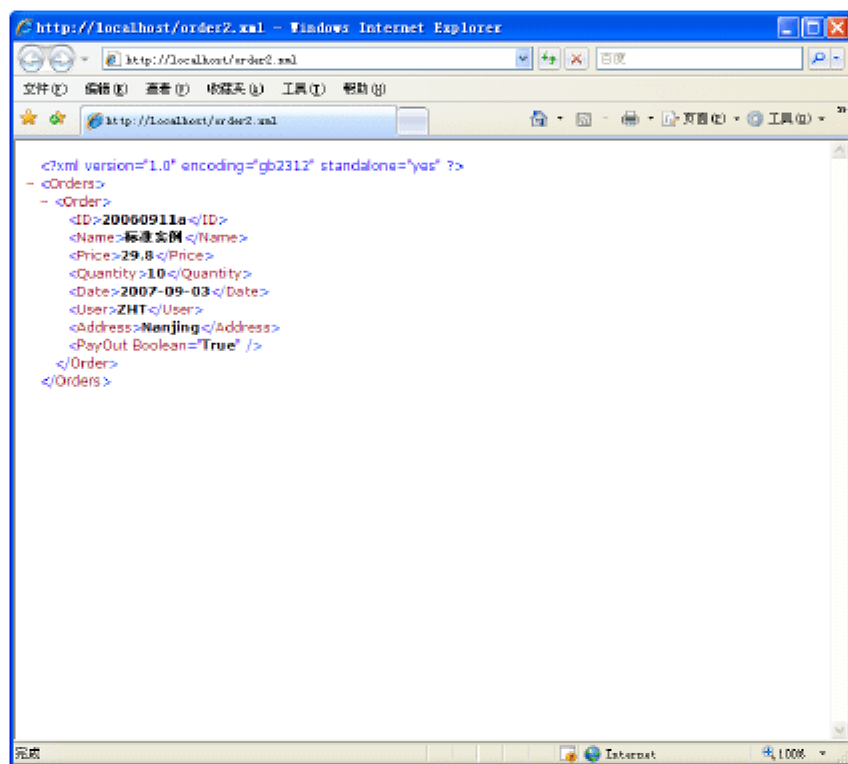


图 11-12　网页浏览效果

11.3　操作 XML 文档

本章前面详细介绍了有关 XML 相关的知识， XML 与 HTML 的区别，XML 的特点，XML 文档的结构，以及如何编写格式正确的 XML 文档等内容。下面将重点说明，在 ASP 中从 XML 文档中提取和处理数据的方法。

11.3.1　文档对象模型

在 ASP 中对 XML 文档进行操作时，首先要掌握 ASP 提供的 XML 对象。使用这些对象，用户可以方便地操作 XML 文档的节点，从而真正地把 XML 实现为数据的载体。

第一个需要掌握的是 XML 的文档对象模型，文档对象模型(Document Object Model,DOM)是针对 HTML 和 XML 文档的 API，定义了文档的逻辑结构以及访问它们的方法。也就是说，它定义了一个标准的访问和处理 XML 结构的方法。XML Document 对象是描述 XML 源文件的顶级对象，它的方法和属性可以产生出其他所有的 XML 对象。DOM 对象对 XML 的描述完全遵循 XML 本身的层次结构，所以只要了解 XML 的文档结构，就很容易了解 DOM 对象对 XML 的描述。

下面通过一个简单的 XML 文档，学习使用 DOM 的方法。

```
<?xml version="1.0" encoding="gb2312" standalone="yes"?>
<Orders>
<Order>
<ID>200708011a</ID>
<Name>标准教程</Name>
<Price>29.8</price>
<Quantity>10</Quantity>
</Order>
<Order>
<ID>200708011b</ID>
<Name>实用教程</Name>
<Price>29.8</price>
<Quantity>25</Quantity>
</Order>
</Orders>
```

使用 DOM 的术语，XML 文档本质上是分层结构的文档。也就是说，总是有一个顶层或根元素，然后在下面有一些子元素。所以上面的文档可以用如图 11-13 所示来表示。

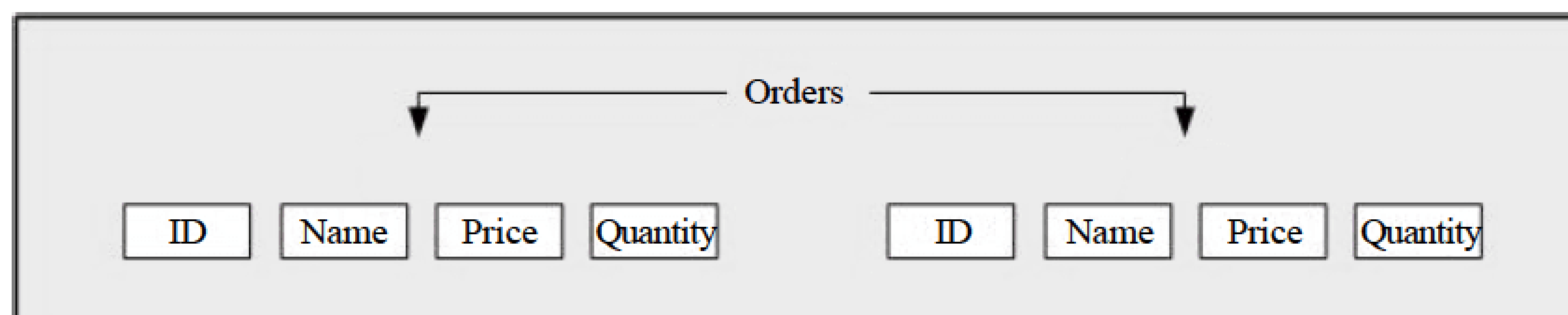


图 11-13　XML 的 DOM 分层结构

如果文档包含的子元素很多，将会得到一个层次很深的树状图。使用 DOM 的术语来说，这些元素也称为节点(Node)，一个节点代表树状结构中的一个普通元素。

1. 基本对象

为了表现树状的分层结构，DOM 中提供了整套的对象、方法以及属性以使用户能处理 DOM。在这里先讨论最常用到的对象和属性。首先介绍对象，常用的有以下 3 种。

- Node 对象：在分层结构中的单个节点。
- NodeList 对象：节点集合。
- NameNodeMap 对象：允许用名字和索引号来访问的节点集合。

除了以上 3 个对象以外，还有一些用来访问节点的属性，如表 11-1 所示。

表 11-1　节点属性及说明

属　　性	说　　明
childNodes	返回一个包含子节点的 NodeList 对象
firstChild	返回当前节点的第一个子节点
NodeName	节点名称
previousSibling	返回上一个兄弟节点，也就是在分层结构中相同层上的前一个节点
lastChild	返回当前节点的最后一个子节点
ParentNode	返回当前节点的父节点
NodeValue	节点的值
nextSibling	返回下一个兄弟节点，也就是在分层结构中相同层上的后一个节点

下面将针对本节前面所示 XML 程序中所用的文档的节点结构进行分析。

在该 XML 文档中，根节点(根元素)Orders 下的子节点存放于 ChildNodes 集合中，而且由于根节点 Orders 只有一个子节点，所以属性 FirstChild 指向了同一个节点 Order。在这种情况下，节点集合中唯一的节点是 ChildNodes(0)。

节点 Order 有 4 个子节点，存放于 ChildNodes 集合中，属性 FirstChild(也就是 ChildNodes(0))指向节点 ID，属性 LastChild(也就是 ChildNodes(3))指向节点 Quantity，而属性 PreviousSibling 和 NextSibling 则在同一层，分别指向节点集合中上一个和下一个兄弟节点。

例如，假设有一个指向 Orders 的名称为 RootNode 的节点对象，则节点的代码表示方法如表 11-2 所示。

表 11-2　节点的代码表示法

代　　码	指 向 的 节 点
RootNode.ChildNode(0)	Order
RootNode.ChildNode(0).FirstChild	ID
RootNode.ChildNode(0).FirstChild.NextSibling	Name
RootNode.ChildNode(0).FirstChild.ParentNode	Order
RootNode.ChildNode(0).FirstChild.NextSibling.ParentNode	Order

2. 特殊 DOM 对象

XML 具有丰富的扩展性能，可以满足文档的多样性。所以针对不同的节点类型，还有一些特殊的对象。这些对象继承了 Node 对象的大部分方法和属性，同时也增强了一些特殊方法与属性，用于处理那些特殊的节点类型。如表 11-3 所示列出了这些对象。

表 11-3　DOM 特殊对象及说明

对　　象	说　　明
DocumentXML	文档的根对象
Element	一个 XML 对象
Attribute 或 Attr	一个 XML 属性
EntityReference	一个实体的引用
Comment	一个 XML 注释元素
CharacterData	文档中用于文本信息的基本对象
Notation	一个符号，等同于 DTD 中的!NOTATION
Text	元素或属性节点的文本内容
Implementation	应用程序特殊的实现细节
ProcessingInstruction	一条处理指令，存放于<??>段中
CDATASection	未经过分析的字符数据，等同于 DTD 中的！CDATA
DocumentType	与 XML 有关的 DTD 或模式的信息，等同于 DTD 中的 DOCTYPE
DocumentFragment	Document 的副本，用于临时存储操作或文档的插入操作
Entity	一个经过分析或未经过分析的实体，等同于 DTD 中的！ENTITY

由于这些 DOM 对象的特殊性，因此一般不会使用到所有这些对象，在这里就不再深入解释它们及其与 XML 的关系。

11.3.2　读取 XML 数据

掌握文档对象模型是操作 XML 文档的必备基础，第 11.3.1 节中已经介绍了这些知识。与使用活动服务器组件一样，在操作 XML 文档对象之前必须先使用 Server.CreateOjbect 创建对象的实例。在 ASP 中创建 DOM 对象的基本语法如下(以下两种方法都可以创建 DOM 对象)：

```
set ObjXML=Server.Createobject("MSXML.DOMDocument")
set ObjXML=Createobject("Microsoft.XMLDOM")
```

创建完这个对象后就可以使用其内部的函数了，下面将通过实例来说明其使用方法(要使用 ASP 读取 XML 文档中的数据，可以参考下面介绍的方法)。

(1) 创建 Microsoft.XMLDOM 对象，设置对象的 ASYNC 属性为 True，这个属性用来

设置打开后的 XML 数据是否与 XML 文件中的数据同步，也可以理解为动态更新的意思。

```
Set objXML=CreateObject("Microsoft.XMLDOM")
objXML.ASYNC=False
```

(2) 利用 Load 函数载入 XML 文档的信息。例如，要读取 XML 文档中 Order 元素(节点)中的数据，就可以使用如下所示的方法。

```
set objlst=objXML.GetElementsByTagName("Order")
```

(3) 还可以使用 Length 属性统计出共有多少个 Order 子节点：

```
intNo0fHeadlines=Objlst.length
```

(4) 用户已经读取一个 XML 文档的信息了，可以使用下面所示的语句将结果输出。

```
Order.xml 文档中 Order 节点共有：<%=intNo0fHeadlines%>个
```

(5) 将以上代码以文件名 ReadxmlData.asp 保存，然后使用浏览器打开查看结果即可显示如图 11-14 所示的结构。

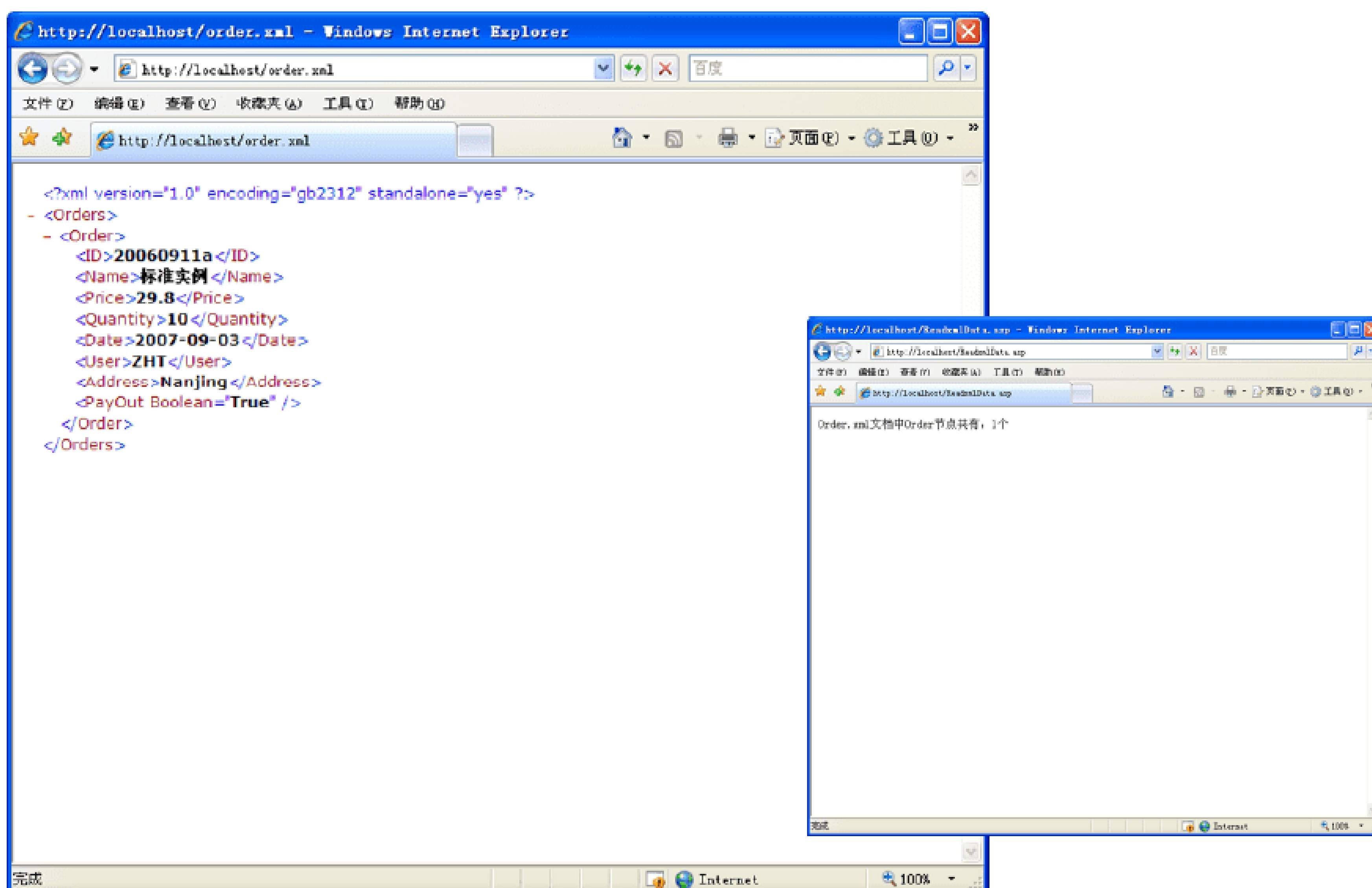


图 11-14　读取并显示 XML 文档的数据

11.3.3　向 XML 输出数据

要操作 XML 文档，还需要使用 XMLDOM 对象向 XML 文档写入信息，下面就将介绍向 XML 文档中写入文件内容的方法。

(1) 要确定操作的 XML 文档的位置，例如在当前文件夹下的 Order.xml 文件：

```
Dim XmlDoc, Flag, FileName
FileName = Server.MapPath("Order.xml")
```

(2) 创建 XML DOM 对象的实例，并载入 XML 文档，设置 ASYNC 属性为 False，即不同步更新：

```
set XmlDoc = Server.CreateObject("Microsoft.XMLDOM")
XmlDoc.ASYNC = False
Flag = XmlDoc.Load(FileName)
```

(3) 为了确保写入 XML 文档的成功，用户需要添加下面的语句判断 XML 对象载入 XML 文档是否成功，成功则执行后面的语句：

```
If Flag Then
```

(4) 成功载入后，先创建一个节点 Order：

```
Dim rootNode, newNode
Set rootNode=XmlDoc.CreateElement("Order")
```

(5) 上面的语句在 XML 文档中的根元素下创建了一个节点<Order>，由于这是一个空字节点，因此还要向这个节点插入内容，如下所示：

```
Set newNode=XmlDoc.CreateElement("ID")
newNode.Text="20060911b"
rootNode.AppendChild(newNode)
```

(6) 第一行语句创建名为 ID 的节点，然后使用其 Text 属性赋值，接下来执行 AppendChild 方法将节点和内容插入到 Order 节点中去。下面使用同样的方法，插入两个节点：

```
Set newNode=XmlDoc.CreateElement("Name")
newNode.Text="标准教程"
rootNode.AppendChild(newNode)
Set newNode=XmlDoc.CreateElement("Price")
newNode.Text-"29.8"
rootNode.AppendChild(newNode)
```

(7) 最后，将对 XML 文档所做的修改保存，可以使用 XmlDoc.Save(文件名)方法：

```
XmlDoc.DocumentElement.AppemdChild(RootNode)
XmlDoc.Save(FileName)
End If
```

(8) 与第 11.3.2 节一样，添加如下的测试成功语句：

```
写入 Order.xml 文档成功
```

(9) 将以上代码以 WritexmlData.asp 为文件名进行保存，并在浏览器中打开，如图 11-15 所示为测试成功语句以及写入后的 XML 文档。

图 11-15　写入并显示 XML 文档的数据

11.4　习　　题

11.4.1　填空题

(1) XML 是基于____________的标记语言。

(2) 一般称包含在元素开始标记和结束标记之间的信息为____________。

(3) 检查 XML 文档格式是否正确的程序叫做____________，它可以分为____________和检验两种类型。

(4) ____________是针对 HTML 和 XML 文档的 API，使用时需要通过 Server.CreateObject("____________")或者 CreateObject("____________")语句实例化对象。

11.4.2　选择题

(1) 在 XML 术语中，将一个标记及其包含的信息称为___________。

A. 元素　　B. 标记块

C. 方法　　D. 标签

(2) 关于 XML 的描述，下列正确的是___________。

A. XML 只定义属性　　B. XML 只定义方法

C. XML 不定义任何元素和属性　D. XML 不使用元素或属性

11.4.3　问答题

(1) 简述在标记语言的发展过程中经过几代语言？

(2) 简述创建 XML 文档的步骤。

(3) 编写 XML 文档需要注意哪些原则？

11.4.4　操作题

(1) 参考【练习 11-2】的操作，创建一个简单的 XML 文档，了解创建 XML 文档的方法。

(2) 以上例创建的 XML 文档为基础，通过 Internet Explorer 浏览器解析创建的 XML 文档。

第12章　ASP程序调试与错误处理

ASP 程序在运行时有可能出现各种错误，这些错误可分为 3 类：语法错误、运行错误和逻辑错误。ASP 程序调试的过程就是发现和改正脚本错误的过程。通过调试，可以找出导致错误的原因并找出发生错误的位置。

通过本章的理论学习和练习，用户应了解和掌握以下内容：

- 掌握调试 ASP 脚本的方法
- 了解 ASP 错误提示
- 了解自定义错误信息的方法
- 了解事务

12.1　调试 ASP 程序脚本

开发程序的过程中难免会发生错误，这些错误有时是在编写程序代码时不小心输入错误，有时是因访问者操作不当而引起的。无论开发人员如何小心谨慎，编写脚本时总会有疏忽大意之处，这样运行时也就可能出现这样或那样的错误，而且随着脚本变得越来越复杂，出错的概率也将成倍增长。因此如何避免错误，找出错误便成了 ASP 程序设计中不可缺少的一环。通过调试，就可以找出导致错误的原因并找出错误发生的位置。

12.1.1　认识 ASP 脚本错误

脚本错误大致分为 3 类，即语法错误、运行错误和逻辑错误。语法错误是一种经常遇到的错误，程序没有按照规定的语法书写就会出现语法错误。这种错误常发生在初学者身上，例如关键词拼写错误，If 没有 Then 与之配对，字符串没有用双引号等，都会引发语法错误。

程序在执行时所发生的错误即为运行错误。例如以 0 作为除数导致程序无法继续执行，如下所示：

```
<% X=1/0%>
```

运行错误通常是开发人员在编程时不够严谨、处理不当而遗留下来的潜在性错误。运行错误必须改正，脚本才会不中断地执行。

逻辑错误是脚本未按预期方式执行所产生的错误，即脚本中的代码本身是合法的，而

且能够执行，但执行结果不正确。例如，服务器端脚本将对列表中的值进行排序，若在脚本中本该用小于号(<)的地方错误地使用了大于号(>)，则会导致排序结果不正确。逻辑错误最麻烦，也最难发现。

【练习 12-1】参考上面所介绍的内容，通过调试下面的 ASP 脚本程序，认识 ASP 脚本在调试时可能会出现的语法错误、运行错误和逻辑错误。

(1) 创建一个名为 Test_1.asp 的文件，其代码如下所示：

```
<%
Option Explicit
Dim intNum,intFact
%>
<HEAD><TITLE>计算数的阶乘</TITLE></HEAD>
<BODY>
<Center><H4>脚本的调试</H4></Center>
<HR>
<FORM ACTION="<%=Request.ServerVariables("SCRIPT_NAME")%>" METHOD="GET">
<P>
计算一个数的阶乘：<P>
请输入 0-10 之间的数：<Input Name="Nums" Size="5">
<P>
<Input Type=Submit Value="提交"><Input Type=Reset Value="重置">
</Form>
<Hr>
<%If Request.ServerVariables("QUERY_STRING")<>"" Then
     intNum=cInt(Trim(Request.QueryString("Nums")))
  If intNum>10 And intNum<0 Then
     Response.Write "输入的数据超出计算范围，请重新输入！"
  Else
     intFact=factorial(intNum)   '调用 Factorial 函数来计算阶乘
     Response.Write intNum &"的阶乘是："&intFact
  End If
End If%>
</Body></Html>
<Script Language=VBScript RunAt=Server>
Function factorial(num)
If num=0 OR num=1 Then  '若为 0 或 1，则其阶乘是 1
factorial=1
Else
factorial=num*factorial(num-1)  '进行递归直至完成
End If
End Function
</Script>
```

(2) 将 Test_1.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 启动 IE 浏览器，在地址栏中输入 http://localhost/Test_1.asp 并按 Enter 键打开网页，效果如图 12-1 所示。

(4) 在文本框中输入 0~10 之间的数字(例如输入 3），然后单击“提交”按钮，将在页面下方得到所输入数字的阶乘，如图 12-2 所示。



图 12-1　脚本调试页面



图 12-2　计算数字的阶乘

(5) 在如图 12-1 所示页面的文本框中输入 0~10 以外的数字（例如输入－3，如图 12-3 左图所示），然后单击“提交”按钮，浏览器将会显示如图 12-3 右图所示的错误，提示用户 ASP 脚本出现错误并显示脚本错误类型——运行错误。



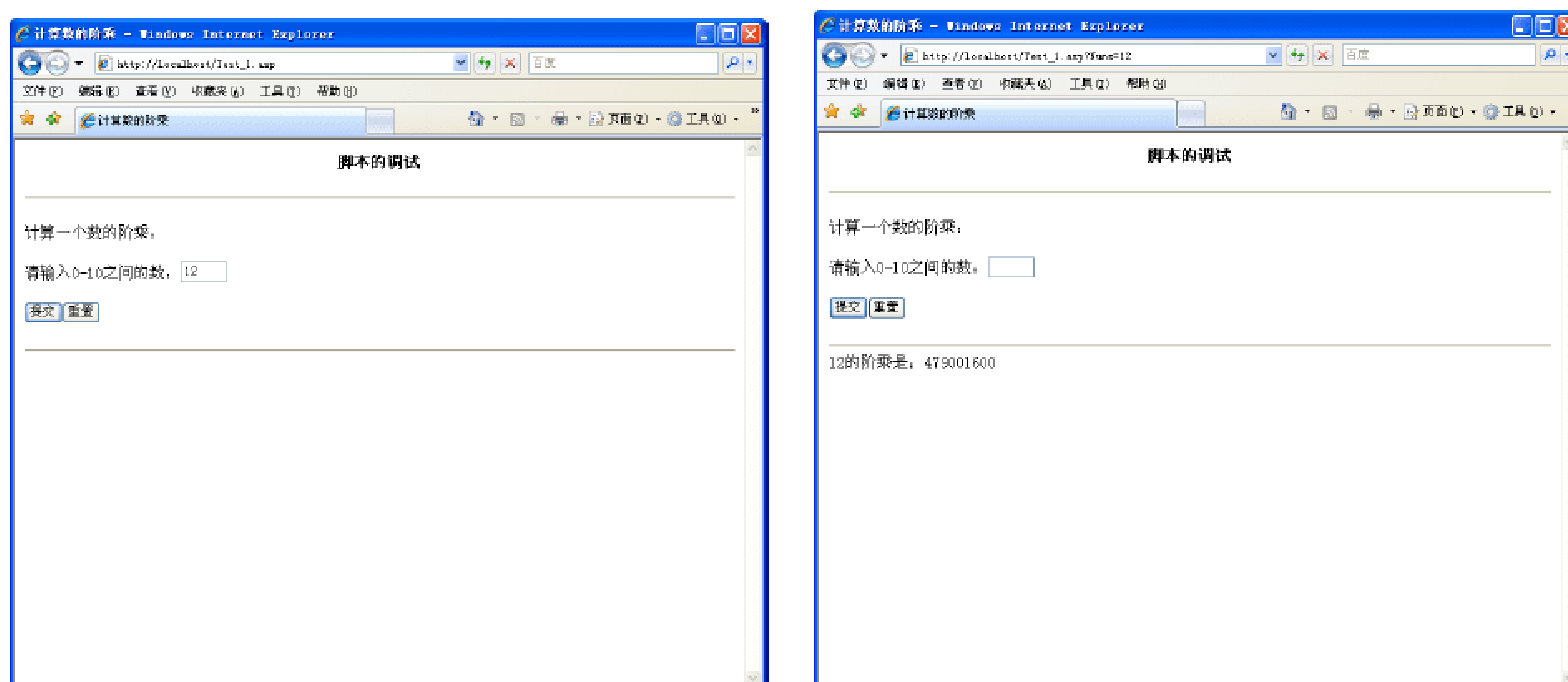
输入“－3”



浏览器提示 ASP 脚本运行错误

图 12-3　测试 ASP 脚本的运行错误

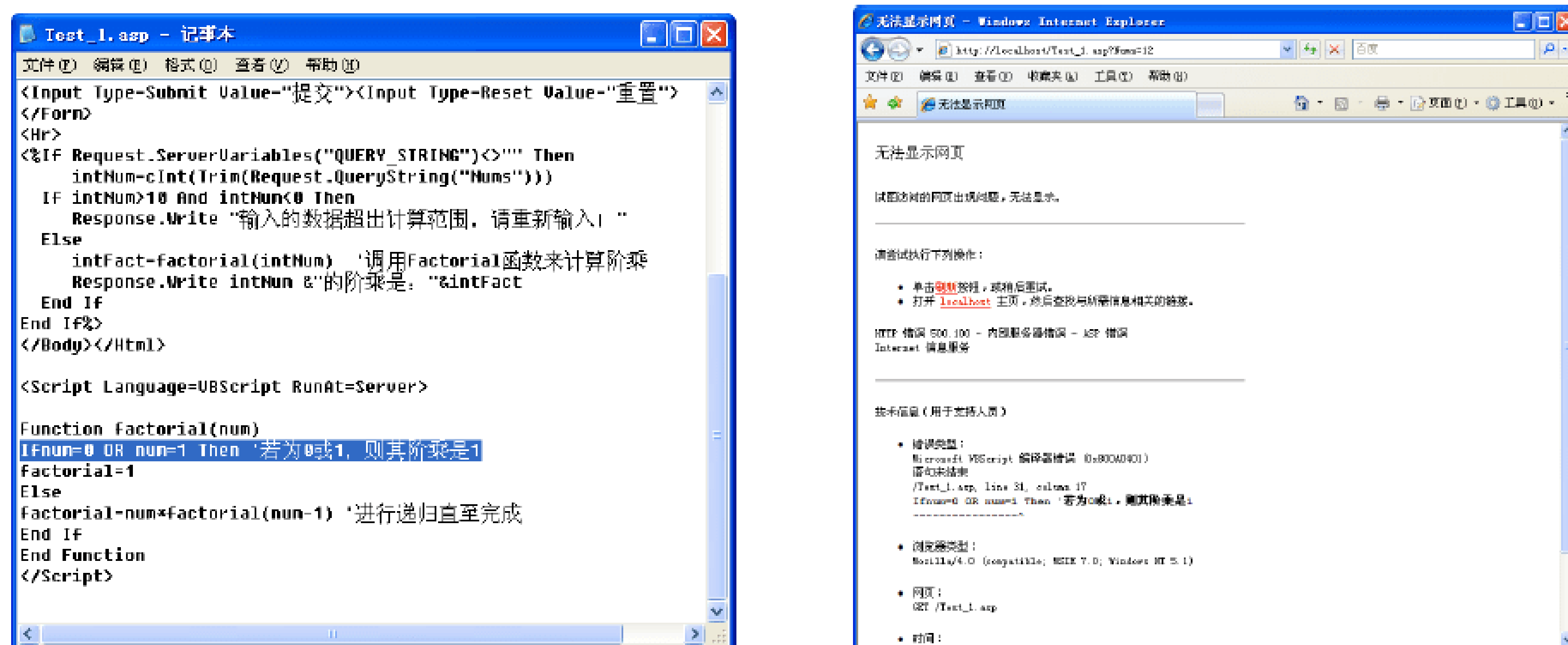
(6) 在如图 12-1 所示页面的文本框中输入 0~10 以外的数字(例如输入 12，如图 12-4 左图所示)，然后单击“提交”按钮，ASP 程序将正常运行，而不会显示“输入的数据超出计算范围，请重新输入！”提示，这是因为 ASP 脚本出现了逻辑错误。



输入 12　　　　　　　　　　ASP 脚本出现逻辑错误

图 12-4　测试 ASP 脚本的逻辑错误

(7) 将记事本中的代码故意写错，如图 12-5 左图所示。完成后将 ASP 脚本保存并在浏览器中运行。这时，浏览器将会提示 ASP 程序出现语法错误，如图 12-5 右图所示。



错误的 ASP 脚本　　　　　　　　　　浏览器提示语法错误

图 12-5　测试 ASP 脚本的语法错误

12.1.2　使用 Microsoft 脚本调试器

IIS 自带一个 Microsoft 脚本编辑器，使用它可查看和编辑 HTML 文件中的 HTML 标记，还可向文件中添加脚本并对其进行调试。Microsoft 脚本编辑器的主要功能有：

- 逐行运行服务器端脚本。
- 在服务器端脚本执行时，打开命令窗口监视变量、属性或数组元素的值。
- 在脚本的指定行设置断点以暂停执行服务器端脚本。
- 运行服务器端脚本时跟踪过程/函数。

用 Microsoft 脚本编辑器调试 ASP 脚本之前，必须先将 Web 服务器配置为“启用 ASP 调试”。

【练习 12-2】在 Internet 信息服务管理器中启用 ASP 调试工具，并打开 Microsoft 脚本编辑器。

(1) 启动 IE 浏览器后，选择“工具”|“Internet 选项”命令（如图 12-6 所示），打开“Internet 选项”对话框。

(2) 在“Internet 选项”对话框中选择“高级”选项卡，在“设置”选项区域中取消选择“禁用脚本调试(Internet Explorer)”和“禁用脚本调试(其他)”复选框(如图 12-7 所示)后，单击“确定”按钮。

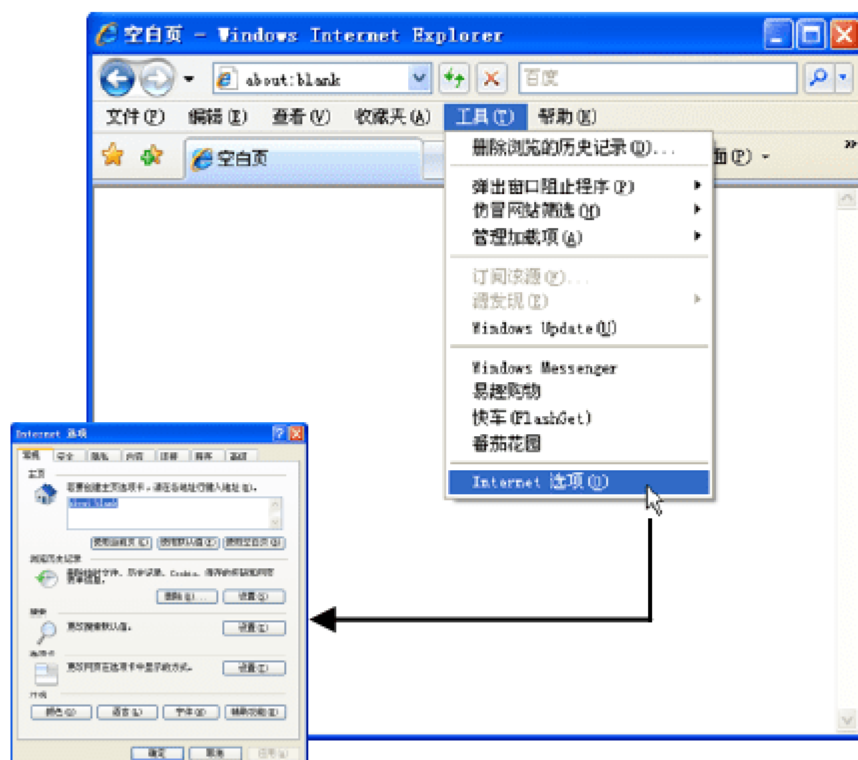


图 12-6　打开“Internet 选项”对话框

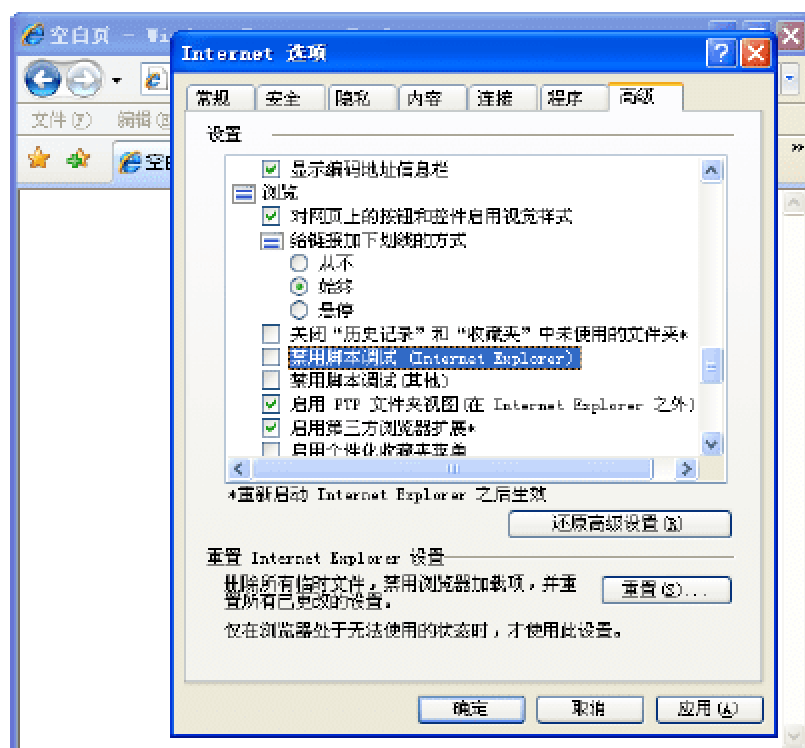


图 12-7　设置“高级”选项卡

(3) 打开“Internet 信息服务”控制台，在 Web 站点或应用程序的起始目录单击鼠标右键，在弹出的快捷菜单中选择“属性”命令，打开“默认网站属性”对话框，如图 12-8 所示。

(4) 选择“主目录”选项卡，单击“配置”按钮，打开“应用程序配置”对话框，如图 12-9 所示。

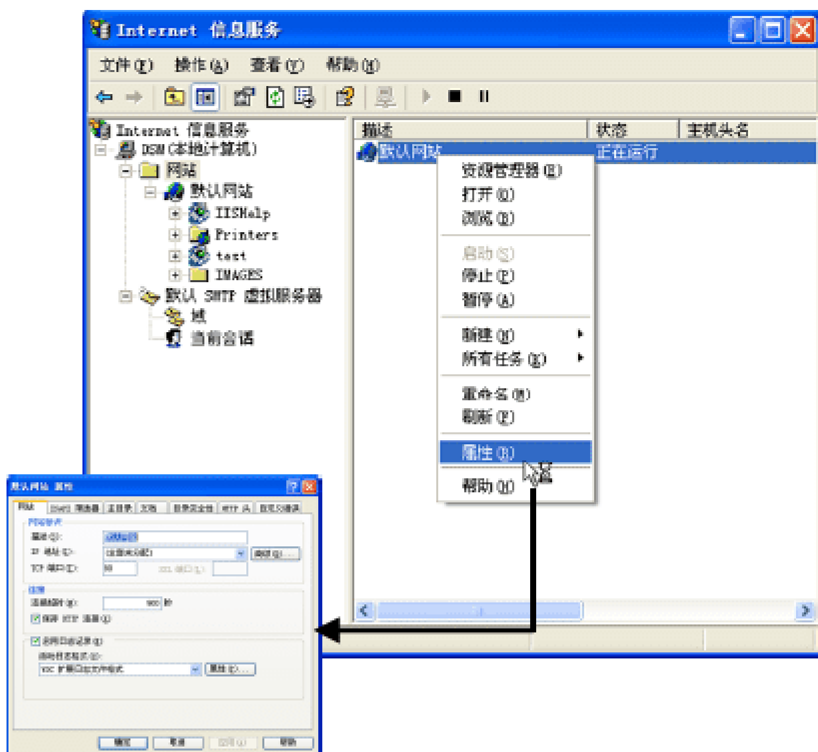


图 12-8　打开“默认网站属性”对话框

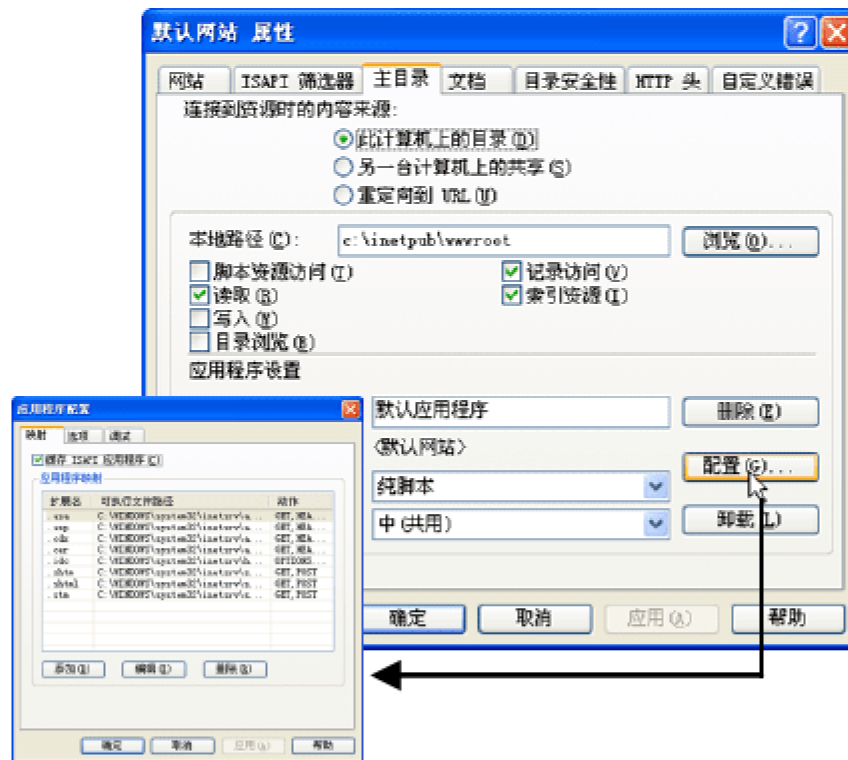


图 12-9　“应用程序配置”对话框

(5) 选择“调试”选项卡，选中“启用 ASP 服务器脚本调试”复选框，如图 12-10 所示。这样当脚本产生错误或 ASP 在脚本中遇到断点时，将启动脚本调试器。

(6) 若选中“向客户端发送详细 ASP 错误消息”单选按钮，则可发送特定调试信息(包括文件名、错误消息和行号)到浏览器；若选中“向客户端发送文本错误消息”单选按钮，则产

生错误时发送默认错误消息到浏览器，特定错误消息被写入错误日志(可在下面的文本框中更改默认错误消息)。

(7) 单击“确定”按钮，保存设置并关闭 Internet 信息服务管理器。

(8) 启用 ASP 调试后，ASP 程序若在运行时发生错误，系统会自动弹出提示对话框，提示是否要选择一种调试器来对程序进行调试，如图 12-11 所示。

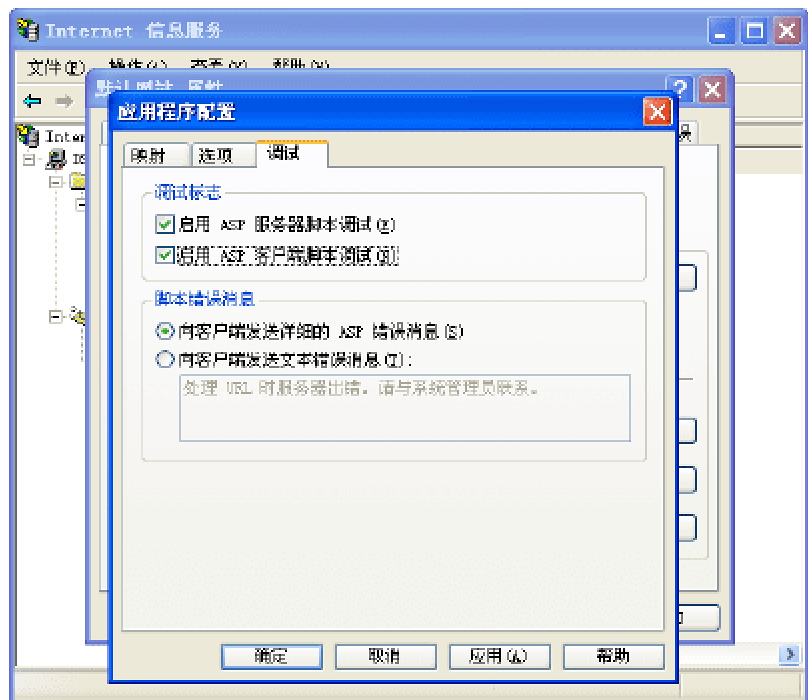


图 12-10　“调试”选项卡

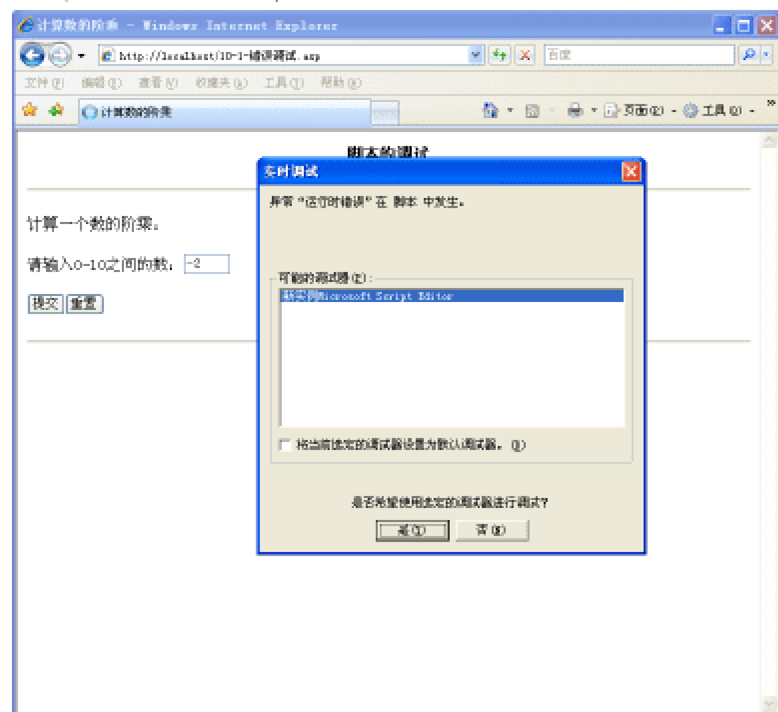


图 12-11　“实时调试”对话框

(9) 在 Microsoft 脚本调试器的“进入并单步执行远程过程调用”对话框（如图 12-12 所示）中单击“确定”按钮，脚本编辑器会打开.asp 文件，将语句指针指向导致错误的行并生成错误消息，如图 12-13 所示。

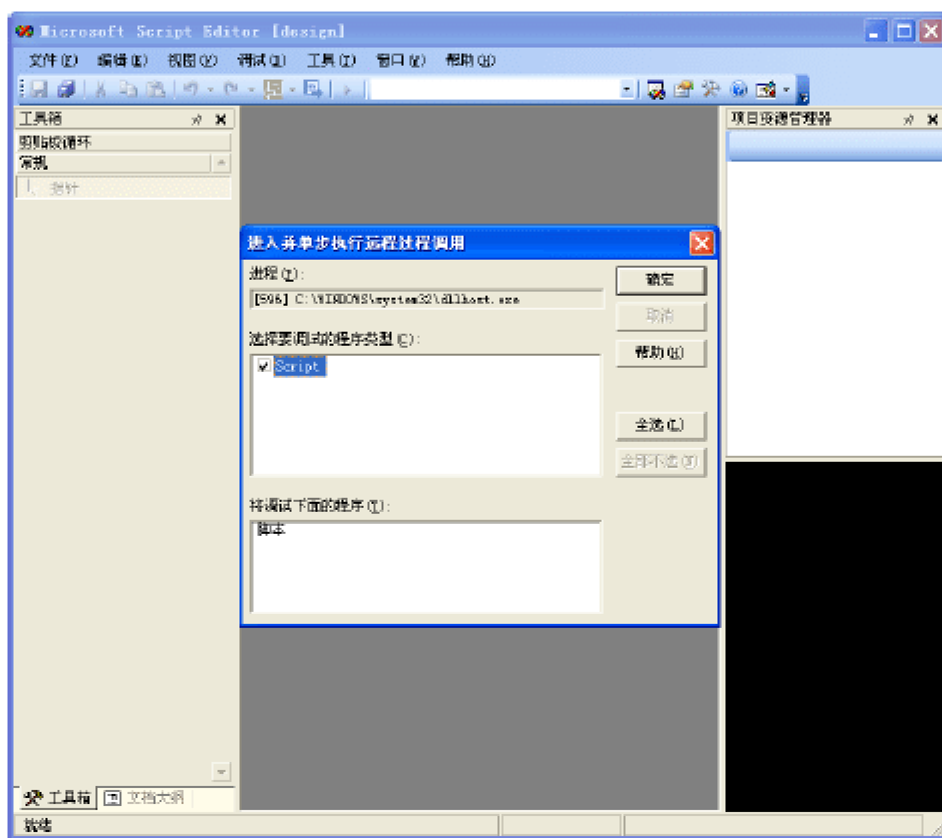


图 12-12　“进入并单步执行远程过程调用”对话框

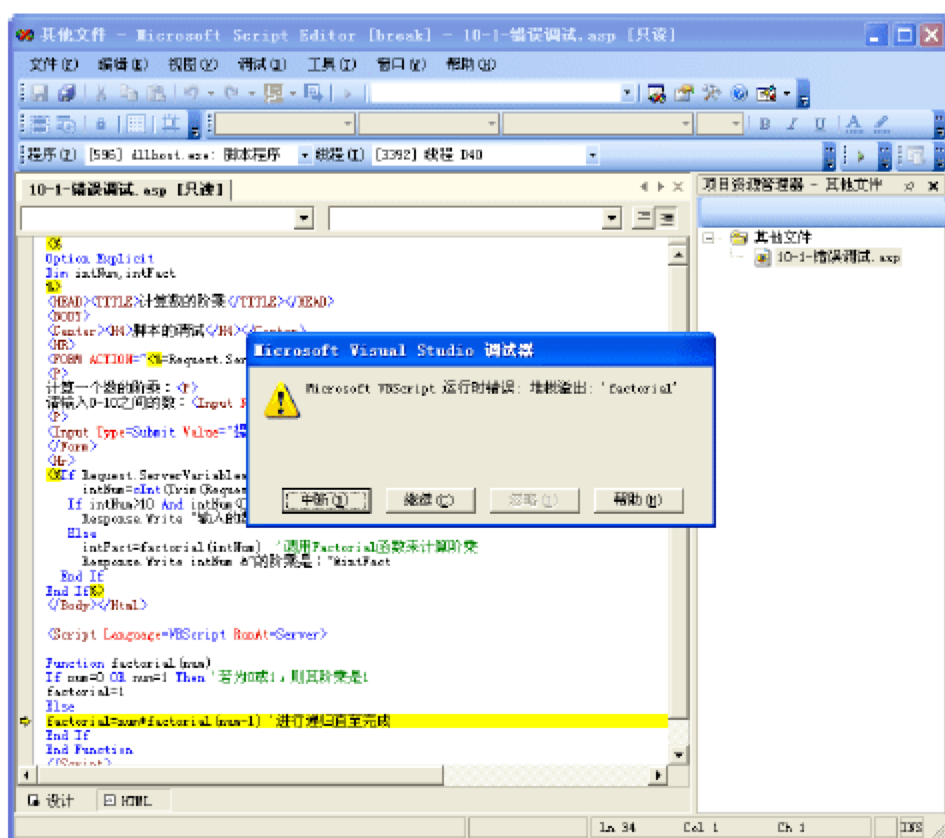


图 12-13　Microsoft 脚本编辑器

注意：

可以使用调试程序查看脚本和定位错误，但不能直接编辑脚本。因为调试器是以只读方式打开脚本的。要改正错误，必须使用编辑程序来编辑脚本，保存更改，然后再次运行该脚本。

12.1.3　断点调试

在 Microsoft 脚本编辑器中，断点调试是一种常用的错误调试技术。断点是在脚本中作了标记的位置，在指定的脚本行暂停执行脚本。

当发生错误但无法轻松地找到错误源时，预先设置“断点”有时会很有用。可以在可疑的行前面设置一个或多个不同的断点，当再次运行程序时，程序在执行到设置了断点的脚本行后，系统会自动启动脚本调试程序显示脚本，并且语句指针指向设置了断点的行。此时可使用调试程序检查脚本中设置的变量或属性的值。改正错误后，可以清除断点，这样脚本就可以不间断地运行了。

【练习 12-3】以【练习 12-1】创建的 ASP 脚本为基础，在程序中设置断点对程序进行断点调试，改正上面程序输入无效数字时脚本没有显示出错提示的逻辑错误。

(1) 根据错误推测检查输入是否有效的代码有问题，在【练习 12-1】创建的 ASP 脚本的 If intNum>10 And intNum<0 Then 语句处设置断点。

(2) 打开 Microsoft 脚本编辑器，选择“调试”|“窗口”|“运行文档”命令，打开“运行文档”窗口，如图 12-14 所示。在窗口中双击一个文件名可打开相应脚本。

(3) 将鼠标指针移到需要设置断点的行(If intNum>10 And intNum<0 Then)，然后单击鼠标右键，在弹出的快捷菜单中选择“插入断点”命令，此时该行左边的空白区会显示断点标记●，如图 12-15 所示。

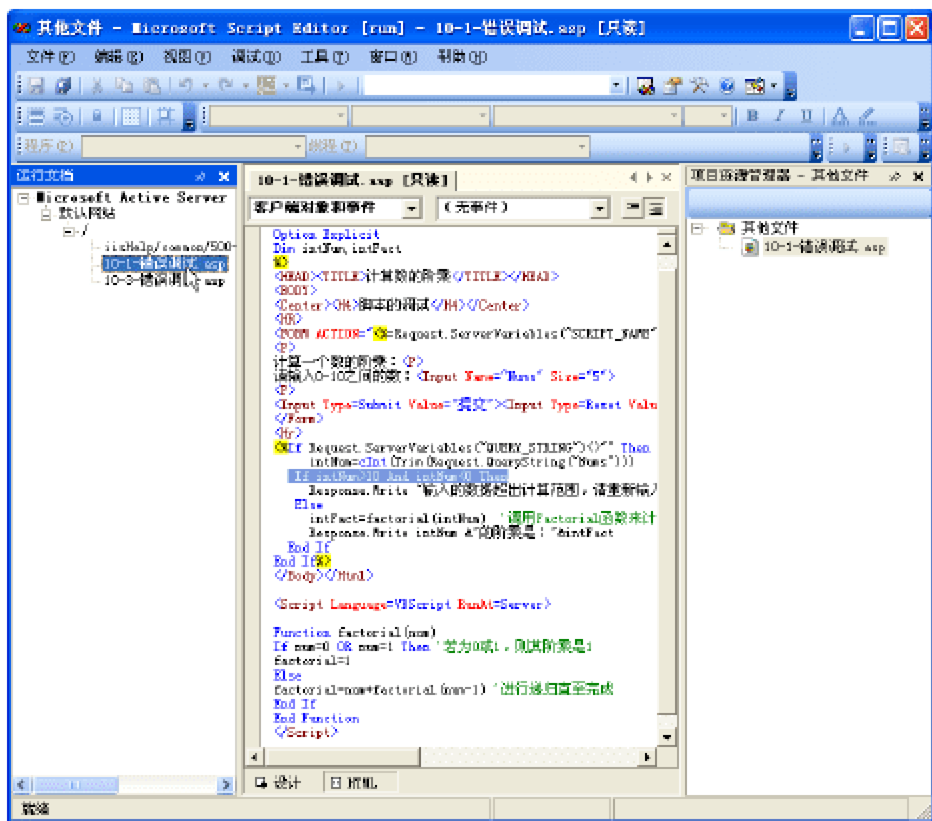


图 12-14　“运行文档”窗口 1

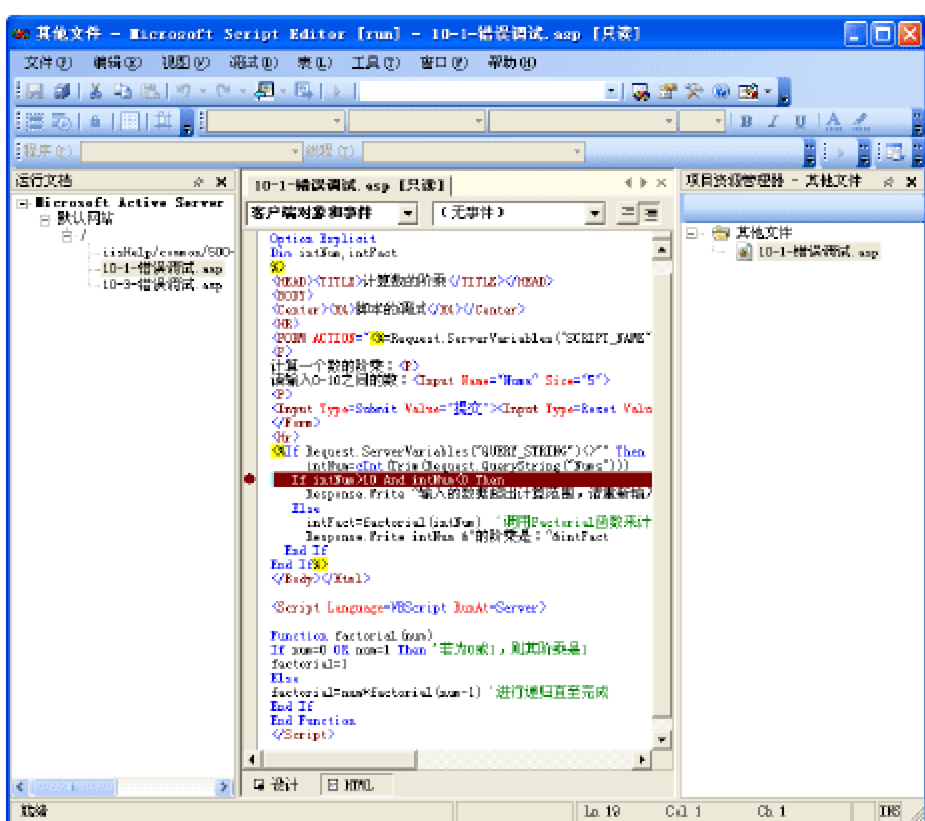


图 12-15　设置断点 1

注意：

只能在至少已运行过的脚本中设置断点。为了能够在第一次调用脚本时暂时中断执行，可以在脚本的可疑行之前插入一个 Stop 语句。执行到 Stop 语句时，脚本会暂停执行。插入 Stop 语句就相当于设置了断点，调试完成后可手工清除所有的 Stop 语句。例如：

```
<%If Request.ServerVariables("QUERY_STRING")<>"" Then
intNum=cInt(Trim(Request.QueryString("Nums")))
```

```
Stop        '暂停脚本执行
If intNum>10 And intNum<0 Then
```

(4) 在浏览器中重新请求脚本，执行到断点所在行后，将在断点处暂停执行并在左边空白行显示断点中止标记。此时，可以在命令窗口中用“？”(问号)来检查变量或属性的值。例如，要检查 intNum 变量的值，可选择“调试”|“窗口”|“即时”命令打开命令窗口，再输入？intNum 并按 Enter 键，如图 12-16 所示。

(5) 选择“调试”|“逐语句”命令，单步执行下面一条语句，本例为 If 语句。此操作执行的结果为，光标将转到 Else 后面的语句处，如图 12-17 所示。

对问题进行分析，发现问题出在 If 语句处。因为当输入不在 0~10 之间的数字时，正常运行应该转到 Then 后的第一条语句处，因此 If 的判断条件可能存在问题。If 中的条件 If intNum>10 And intNum<0 Then 是错误的，因为无论输入什么样的数字，该条件总是 False。应在脚本编辑器中将此条件改为 If intNum>10 Or intNum<0 Then。

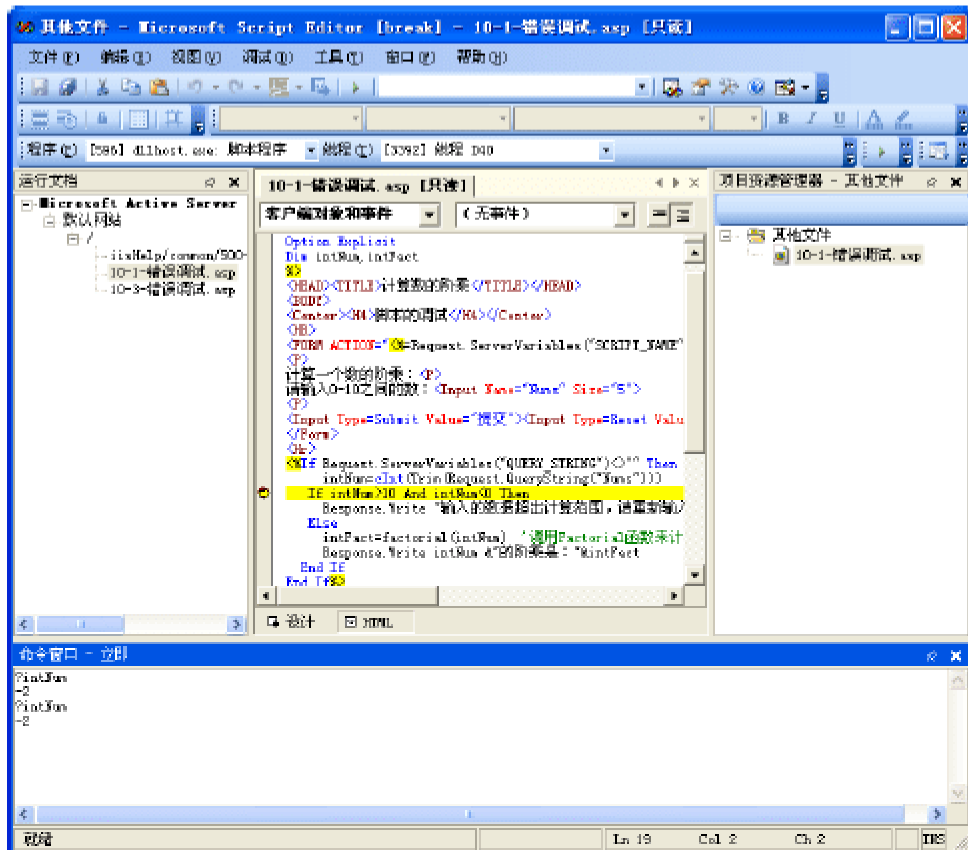


图 12-16　“运行文档”窗口 2

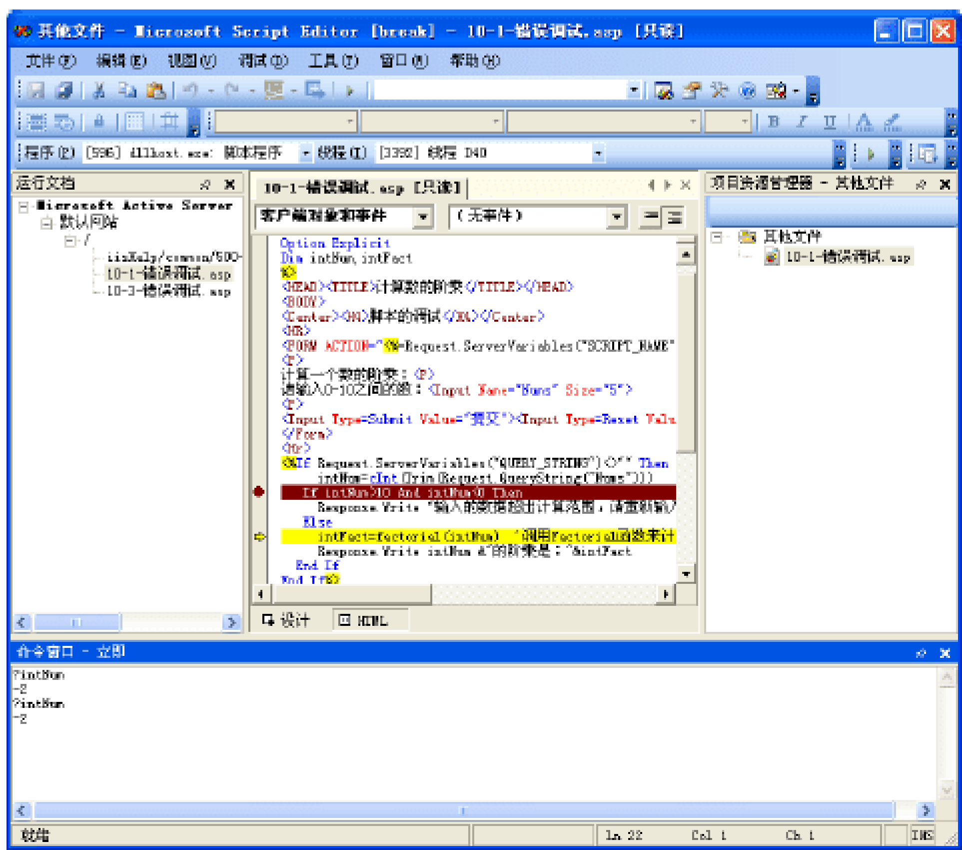


图 12-17　设置断点 2

(6) 选择“调试”|“逐过程”命令，可将下面过程/函数作为一个整体来执行，本例为 factorial()函数。此操作执行的结果为，光标将转到函数调用后的语句处。

(7) 选择“调试”|“清除所有断点”命令，清除程序中设置的所有断点。

12.2　认识 ASP 错误提示

在 ASP 程序运行过程中，可能会因出错而遇到各种错误提示。了解这些错误提示所代表的含义，会方便对程序进行调试。

12.2.1　VBScript 语法错误

如果用户在使用 VBScript 语句编写 ASP 应用程序时，VBScript 语句结构违反了一个

或多个 VBScript 脚本语言语法规则，那么当运行 ASP 脚本时，浏览器就会出现提示 VBScript 语法错误，如图 12-18 所示。

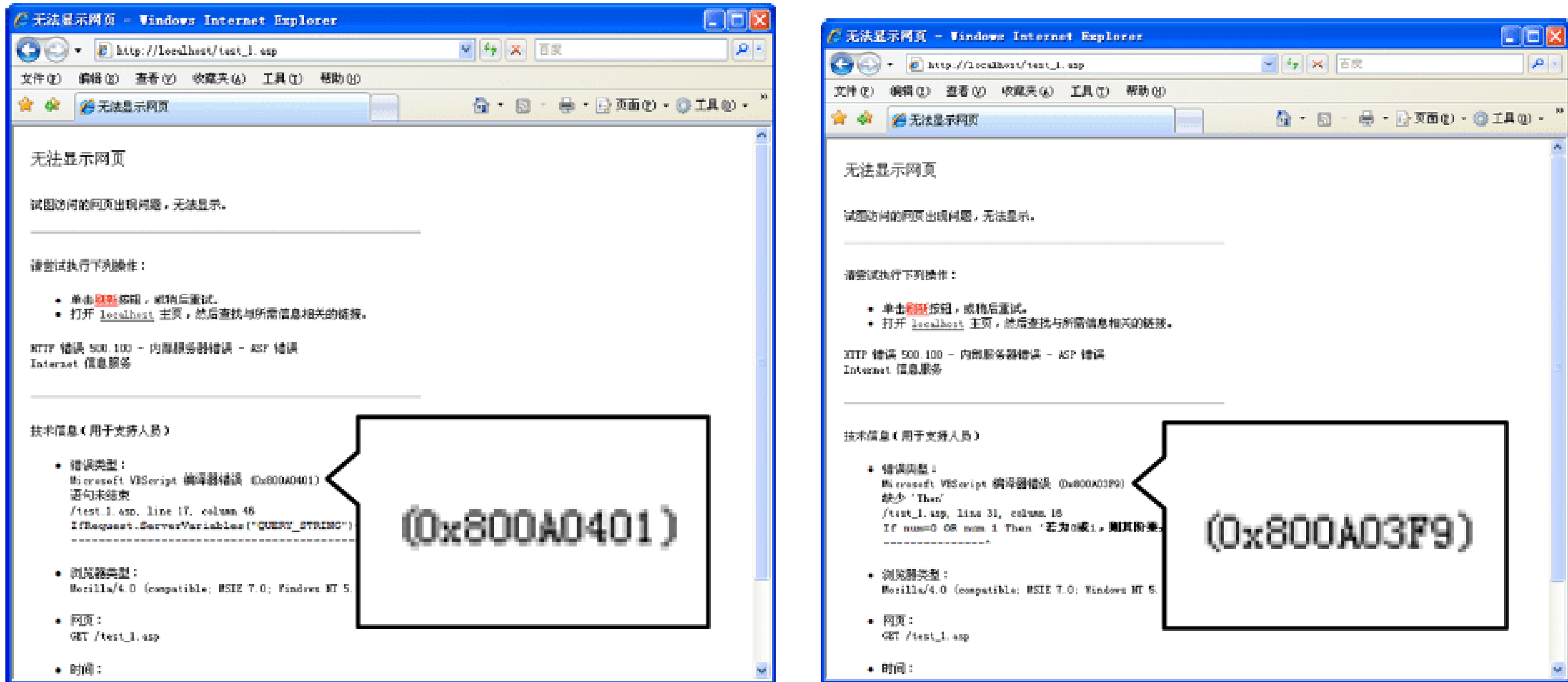


图 12-18　VBScript 语法错误

VBScript 语法错误通常在执行程序前，编译程序时产生。常见的 53 个语法错误及其相关说明如表 12-1 所示。

表 12-1　VBScript 语法错误

错 误 编 号		说　明
十进制	十六进制	
1001	800A03E9	内存不足
1002	800A03EA	语法错误
1003	800A03EB	缺少“：”
1005	800A03ED	需要“(”
1006	800A03EE	需要“)”
1007	800A03EF	缺少“]”
1010	800A03F2	需要标识符
1011	800A03F3	需要=
1012	800A03F4	需要 If
1013	800A03F5	需要 To
1014	800A03F6	需要 End
1015	800A03F7	需要 Function
1016	800A03F8	需要 Sub
1017	800A03F9	需要 Then
1018	800A03FA	需要 Wend
1019	800A03FB	需要 Loop
1020	800A03FC	需要 Next

(续表)

错误编号		说　明
十进制	十六进制	
1021	800A03FD	需要 Case
1022	800A03FE	需要 Select
1023	800A03FF	需要表达式
1024	800A0400	需要语句
1025	800A0401	需要语句的结束
1026	800A0402	需要整数常数
1027	800A0403	需要 While 或 Until
1028	800A0404	需要'While、Until 或语句未结束
1029	800A0405	需要 With
1030	800A0406	标识符太长
1031	800A0407	无效的数
1032	800A0408	无效的字符
1033	800A0409	未结束的串常量
1034	800A040A	未结束的注释
1037	800A040D	无效使用关键字 Me
1038	800A040E	loop 没有 do
1039	800A040F	无效 exit 语句
1040	800A0410	无效 for 循环控制变量
1041	800A0411	名称重定义
1042	800A0412	必须为行的第一个语句
1043	800A0413	不能赋给非 Byval 参数
1044	800A0414	调用 Sub 时不能使用圆括号
1045	800A0415	需要文字常数
1046	800A0416	需要 In
1047	800A0417	需要 Class
1048	800A0418	必须在一个类的内部定义
1049	800A0419	在属性声明中需要 Let,Set 或 Get
1050	800A041A	需要 Property
1051	800A041B	参数数目必须与属性说明一致
1052	800A041C	在类中不能有多个默认的属性/方法
1053	800A041D	类初始化或终止不能带参数
1054	800A041E	PropertyLet 或 Set 至少应该有一个参数

(续表)

错误编号		说　明
十进制	十六进制	
1055	800A041F	不需要 Next
1056	800A0420	只能在 Property 或 Function 或 Sub 上指定 Default
1057	800A0421	说明 Default 时必须同时说明 Public
1058	800A0422	只能在 PropertyGet 中指定 Default

12.2.2　VBScript 运行错误

如果编写 ASP 应用程序的 VBScript 脚本执行系统无法实施的操作，ASP 页面在运行时则会在浏览器中提示 VBScript 运行错误。只有在运行脚本，为变量表达式赋值或分配内存时，才会产生 VBScript 运行错误。常见的 65 个运行错误及其相关说明如表 12-2 所示。

表 12-2　VBScript 运行错误

错误编号		说　明
十进制	十六进制	
5	800A0005	无效过程调用或参数
6	800A0006	溢出
7	800A0007	内存不足
9	800A0009	下标越界
10	800A000A	该数组为定长的或临时被锁定
11	800A000B	被零除
13	800A000D	类型不匹配
14	800A000E	字符串空间溢出
17	800A0011	无法执行请求的操作
28	800A001C	堆栈溢出
35	800A0023	未定义 Sub 或 Function
48	800A0030	加载 DLL 错误
51	800A0033	内部错误
52	800A0034	坏文件名或数
53	800A0035	文件未找到
54	800A0036	坏文件模式
55	800A0037	文件已经打开

(续表)

错误编号		说　明
十进制	十六进制	
57	800A0039	设备 I/O 错误
58	800A003A	文件已经存在
61	800A003D	磁盘空间已满
62	800A003E	输入超出文件尾
67	800A0043	文件太多
68	800A0044	设备不可用
70	800A0046	权限禁用
71	800A0047	磁盘未准备好
74	800A004A	不能用不同的驱动器重新命名
75	800A004B	路径/文件访问错误
76	800A004C	路径未找到
91	800A005B	未设置对象变量
92	800A005C	For 循环未初始化
94	800A005E	非法使用 Null
322	800A0142	不能建立所需临时文件
424	800A01A8	需要对象
429	800A01AD	ActiveX 部件无法创建对象
430	800A01AE	类不支持自动化
432	800A01B0	在自动化操作中未找到文件名或类名
438	800A01B6	对象不支持该属性或方法
440	800A01B8	Automation 错误
445	800A01BD	对象不支持此操作
446	800A01BE	对象不支持指定的参数
447	800A01BF	对象不支持当前的区域设置
448	800A01C0	未找到命名参数
449	800A01C1	参数不可选
450	800A01C2	错误的参数个数或无效的参数属性值
451	800A01C3	对象不是一个集合
453	800A01C5	指定的 dll 函数未找到
455	800A01C7	代码源锁错误
457	800A01C9	这个键已经是本集合的一个元素关联
458	800A01CA	变量使用了一个 VBScript 中不支持的自动化(Automation)类型
462	800A01CE	远程服务器不存在或不能访问

(续表)

错误编号		说　　明
十进制	十六进制	
481	800A01E1	无效图片
500	800A01F4	变量未定义
501	800A01F5	违法的分配
502	800A01F6	脚本对象不安全
503	800A01F7	对象不能安全初始化
504	800A01F8	对象不能安全创建
505	800A01F9	无效的或不合格的引用
506	800A01FA	类未被定义
507	800A01FB	发生异常
5016	800A1398	需要正则表达式对象
5017	800A1399	正则表达式中的语法错误
5018	800A139A	错误的数量词
5019	800A139B	在正则表达式中需要“]”
5020	800A139C	在正则表达式中需要“)”
5021	800A139D	字符集越界
32811	800A802B	元素未找到

12.2.3　ADO 错误

除了在 Error 对象和 Errors 集合中说明的提供者错误之外，ADO 本身也将错误返回到运行时环境的异常处理机制中。

使用编程语言的错误捕获机制(如 Microsoft Visual Basic 中的 OnError 语句)可捕获及处理 ADO 错误，其代码值及相关说明如表 12-3 所示。

表 12-3　ADO 错误代码值

常　量　名	代　　码	说　　明
adErrProviderfailed	3000	提供者未能完成请求的动作
adErrInvalidArgument	3001	应用程序正使用类型错误、超出可接受范围或与其他参数冲突的参数
adErrOpeningFile	3002	在打开所请求文件的同时出现一个错误
adErrReadFile	3003	从指定文件读取时存在一个错误
adErrWriteFile	3004	在写文件的同时存在一个错误
adErrNoCurrentRecord	3021	EOF 或 BOF 为 True，或当前记录已被删除。应用程序所请求的操作需要一个当前记录

(续表)

常　量　名	代　　码	说　　明
adErrIllegalOperation	3219	在这个文本中不允许应用程序请求的操作
adErrCantChangeProvider	3220	提供者不能更改
adErrInTransaction	3246	应用程序在一个事务处理的中途不能明确地关闭一个 Connection 对象
adErrFeatureNotAvailable	3251	提供者不支持应用程序所需的操作
adErrItemNotFound	3265	ADO 不能找到集合中的对象
adErrObjectInCollection	3367	不能附加。对象已经在集合中
adErrObjectNotSet	3420	应用程序所引用的对象不再指向一个合法对象
adErrDataConversion	3421	应用程序正使用当前应用程序不支持的值
adErrObjectClosed	3704	如果此对象关闭，不允许应用程序响应所请求的操作
adErrObjectOpen	3705	如果对象是打开的，则不允许应用程序所请求的这个操作
adErrProviderNotFound	3706	ADO 找不到指定的提供者
adErrBoundToCommand	3707	应用程序无法用 Command 对象将 Recordset 对象的 ActiveConnection 属性更改为它的源数据
adErrInvalidParamInfo	3708	应用程序不正确地定义了一个 Parameter 对象
adErrInvalidConnection	3709	应用程序请求在一个涉及关闭的或非法的 Connection 对象上操作
adErrErrNotReentrant	3710	此操作可重入
adErrStillExecuting	3711	操作仍然在执行
adErrOperationCancelled	3712	这个操作被取消
adErrStillConnecting	3713	操作仍然处于连接中
adErrInvalidTransaction	3714	此事务处理是非法的
adErrErrNotExecuting	3715	此操作没有执行
adErrUnsafeOperation	3716	此操作在这些情况下是不安全的
adErrSecurityDialog	3717	此操作导致显示一个安全对话
adErrSecurityDialogHeader	3718	此操作导致显示一个安全对话标题
adErrIntegrityViolatin	3719	活动由于违反了数据完整性而失败
adErrPermissionDenied	3720	因为您没有权限完成这个操作，所以活动失败
adErrDataOverflow	3721	数据对所提供的数据类型太大
adErrSchemaViolation	3722	动作违反模式
adErrSignMismatch	3723	表达式包含失配的符号
adErrErrCantConvertvalue	3724	值不能转换
adErrErrCantCreate	3725	资源不能建立

(续表)

常　量　名	代　　码	说　　明
adErrColumnNotOnThisRow	3726	这个行上指定的列不存在
adErrURLDoesNotExist	3727	此 URL 不存在
adErrTreePermissionDenied	3728	没有查看此目录树的权限
adErrInvalidURL	3729	所提供的 URL 是非法的
adErrResourceLocked	3730	资源被锁定
adErrResourceExists	3731	资源已经存在
adErrCannotComplete	3732	活动不能完成
adErrVolumeNotFound	3733	文件列未找到
adErrOutOfSpace	3734	因为不能获得足够的空间完成此操作，此操作失败
adErrResourceOutOfScope	3735	资源超出范围
adErrUnavailable	3736	命令不可用
adErrURLamerowDoesNotExist	3737	指定行中此 URL 不存在
adErrDelResOutOfScope	3738	此资源不能删除，因为它超出了允许的范围
adErrPropInvalidColumn	3739	此属性对所选择的列是无效的
adErrPropInvalidOption	3740	为此属性提供了一个无效的选项
adErrPropInvalidValue	3741	为此属性提供了一个无效的值
adErrPropConflicting	3742	设置此属性引起与其他属性的冲突
adErrPropNotAllSettable	3743	并非所有属性都可以设置
adErrPropNotSet	3744	此属性未设置
adErrPropNotSettable	3745	此属性不能设置
adErrPropNotSupported	3746	不支持此属性
adErrCatalogNotSet	3747	活动不能完成，因为目录未设置
adErrCantChangeConnection	3748	连接不能更改
adErrFieldsUpdateFailed	3749	Fields 集合的 Update 方法失败
adErrDenyNotSupported	3750	不能设置 Deny 权限，因为提供者不支持它们
adErrDenyTypeNotSupported	3751	提供者不支持所需 Deny 的类型

12.2.4　ASP 错误代码

ASP 应用程序在运行时的常见错误代码及其相关说明如表 12-4 所示。用户在调试 ASP 脚本程序时，可以参考表中所示的内容识别 ASP 脚本错误。

表 12-4　ASP 错误代码

错误代码	错误消息	说　　明
ASP0100	Out of memory	内存不足(不能分配要求的内存)
ASP0101	Unexpected error	意外错误
ASP0102	Expecting string input	缺少字符串输入
ASP0103	Expecting numeric input	缺少数字输入
ASP0104	Opration not allowed	操作不允许
ASP0105	Index out of ange	索引超出范围(一个数组索引超界)
ASP0106	Type Mismatch	类型不匹配(遇到的数据类型不能被处理)
ASP0107	Stack Overflow	栈溢出(正在处理的数据超出了允许的范围)
ASP0115	Unexpected error	意外错误(外部对象出现可捕获的 exception_name 错误，脚本不能继续运行)
ASP0177	Server.CreateObject Failed	服务器创建对象失败(无效的 progid)
ASP0190	Unexpected error	意外错误(当释放外部对象，产生可捕获的错误)
ASP0191	Unexpected error	意外错误(在外部对象的 OnStartPage 方法中产生可捕获的错误)
ASP0192	Unexpected error	意外错误(在外部对象的 OnEndPage 方法中产生可捕获的错误)
ASP0193	OnStartPage Failed	在外部对象的 OnStartPage 方法中产生错误
ASP0194	OnEndPage Failed	在外部对象的 OnEndPage 方法中产生错误
ASP0240	Script Engine Exception	脚本引擎从 object_name 对象中抛出 exception_name 异常
ASP0241	CreateObject Exception object_name	对象的 CreatObject 方法引起了 exception_name 异常
ASP0242	Query OnStartPage nterface	查询对象 Object_name 的 OnsException

12.3　自定义 ASP 错误信息

当用户试图连接到 Web 站点时，若出现了 HTTP 错误，服务器就会向用户返回一条错误消息，简述试图建立连接时所发生的事情。例如，如果用户试图连接到某个 Web 站点，而该站点已达到最大连接数时，服务器就会以 HTML 页形式返回一条 HTTP 错误，其中包含“用户太多”这样的错误信息。

可以在 IIS 管理单元中配置 IIS 来发送自定义错误信息，而不是发送默认的 HTTP 1.1 错误信息。这些自定义错误信息可以映射为文件名或 URL。可用 IIS 管理单元自定义的 HTTP 1.1 的错误信息如表 12-5 所示。

表 12-5　可以用 IIS 管理单元自定义的 HTTP 1.1 错误消息

错 误 代 码	错 误 消 息	错 误 代 码	错 误 消 息
400	不正确的请求	403.14	目录列表被拒绝
401.1	登录失败	403.15	已超过客户访问许可协议
401.2	由于服务器配置导致登录失败	403.16	客户证书不被信任或无效
401.3	由于资源上的 ACL 导致未被授权	403.17	客户证书过期或者已经无效
401.4	由于筛选器导致授权失败	404	找不到
401.5	由于 ISAPI/CGI 应用程序导致授权失败	404.1	找不到站点
403.1	“执行”访问被禁止	405	不允许的方法
403.2	“读取”访问被禁止	406	不接受
403.3	“写入”访问被禁止	407	需要代理验证
403.4	需要 SSL	412	预期的失败
403.5	需要 SSL128	414	要求的 URI 太长
403.6	IP 地址被拒绝	500	内部服务器错误
403.7	需要客户证书	500.12	正重新启动应用程序
403.8	站点访问被拒绝	500.13	服务器太忙
403.9	用户太多	500.15	不允许请求 Global.asa
403.10	无效配置	500-100.asp	ASP 错误
403.11	密码更改	501	没有执行
403.12	匹配器被拒绝访问	502	错误的网关
403.13	客户证书已被撤销	403.13	客户证书已被撤销

自定义错误消息作为列表显示在 IIS 的管理单元中，IIS 将此管理单元看作单个属性。例如，在 Web 站点级配置一组自定义错误消息后，该服务器下的所有目录都将继承整个自定义错误消息列表。也就是说，两个自定义错误消息列表(对于服务器和目录)并不合二为一。

如果 Web 服务器超过了 CPU 进程限制，就会向客户端发送一条 503.1 自定义的 HTTP 错误。这条错误消息向客户端传递这样的信息，由于 Web 服务器超出这些设置限制，已被暂停或停止。

404.1 错误只会出现在具有多个 IP 地址的计算机上。如果在特定 IP 地址/端口组合上收到客户请求，而且在特定的端口上 IP 地址并没有设置为侦听，则 IIS 将返回 404.1 HTTP 错误。例如，如果一台计算机有两个 IP 地址，而只将其中一个 IP 地址配置为在端口 80 上侦听，则其他 IP 地址从端口 80 收到的任何请求都将导致 IIS 返回 404.1 错误。在服务级设置这一错误，只有当服务器上使用多个 IP 地址时才返回给客户。

12.3.1　添加自定义错误消息

自定义错误消息可分为两种形式：映射为文件或映射为 URL。这两种形式都可通过设置“Internet 信息服务”控制台中的“自定义错误”属性页来实现。

【练习 12-4】在“Internet 信息服务”控制台中添加自定义错误。

(1) 创建包含自定义错误消息的文件并将其放入某个目录中。

(2) 在“Internet 信息服务”控制台中，在欲自定义 HTTP 错误的 Web 站点、虚拟目录、目录或文件上右击，在弹出的快捷菜单中选择“属性”命令，如图 12-19 所示。

(3) 系统弹出属性对话框，选择“自定义错误”选项卡，选中想要更改的 HTTP 错误，如图 12-20 所示。

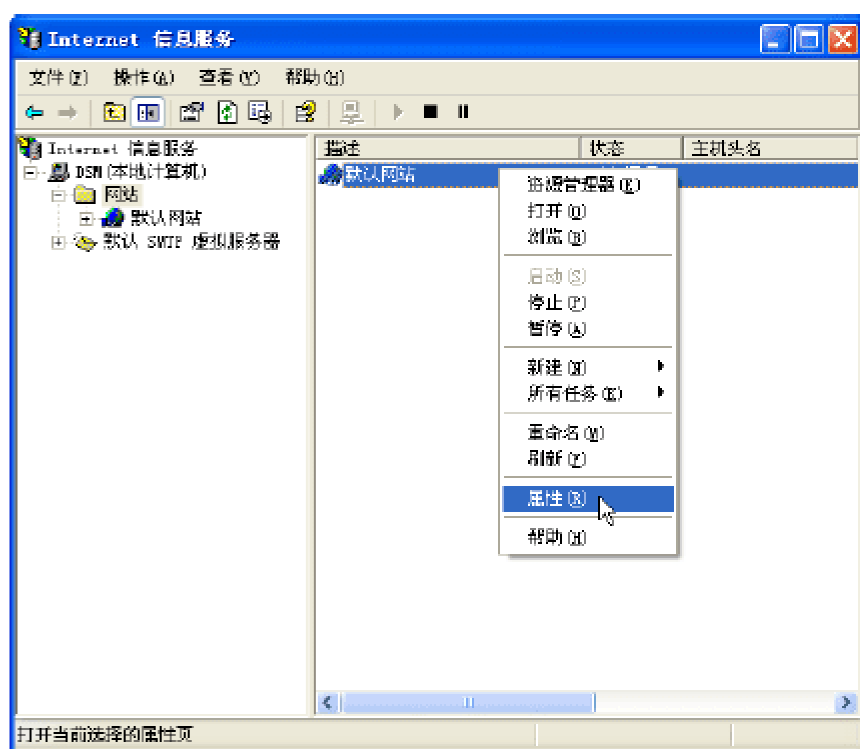


图 12-19　“Internet 信息服务”控制台

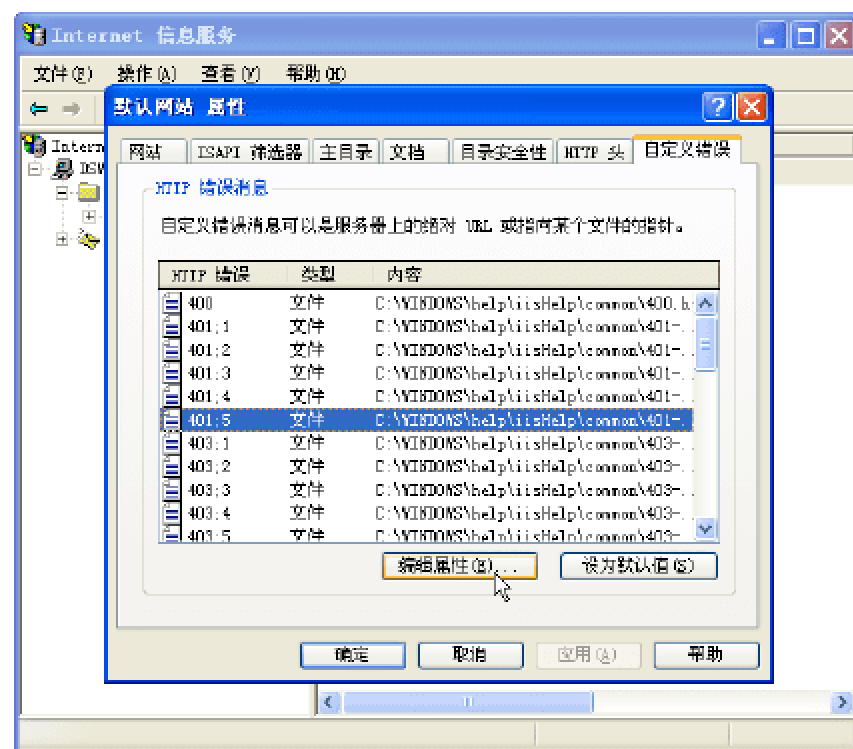


图 12-20　“自定义错误”选项卡

(4) 单击“编辑属性”按钮，系统弹出“错误映射属性”对话框，如果要将错误消息映射为文件，可在“消息类型”下拉列表框中选择“文件”选项(如图 12-21 所示)，然后在“文件”文本框中输入指向自定义错误消息的路径和文件名，或者单击“浏览”按钮来定位硬盘上的文件；如果要映射为 URL，可在“消息类型”框中选择 URL 选项，然后输入指向自定义错误消息的 URL，如图 12-22 所示。

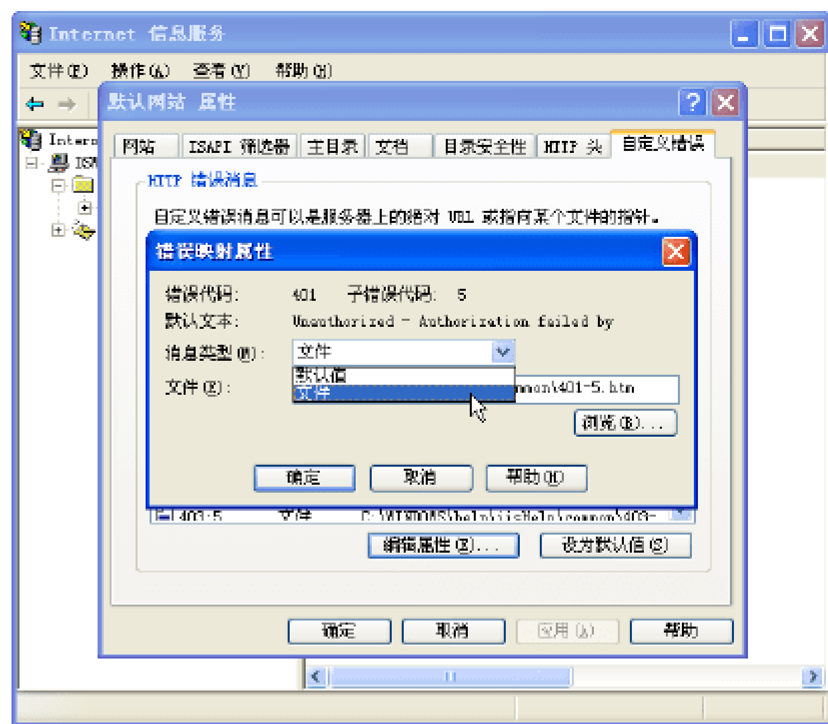


图 12-21　选择“文件”选项

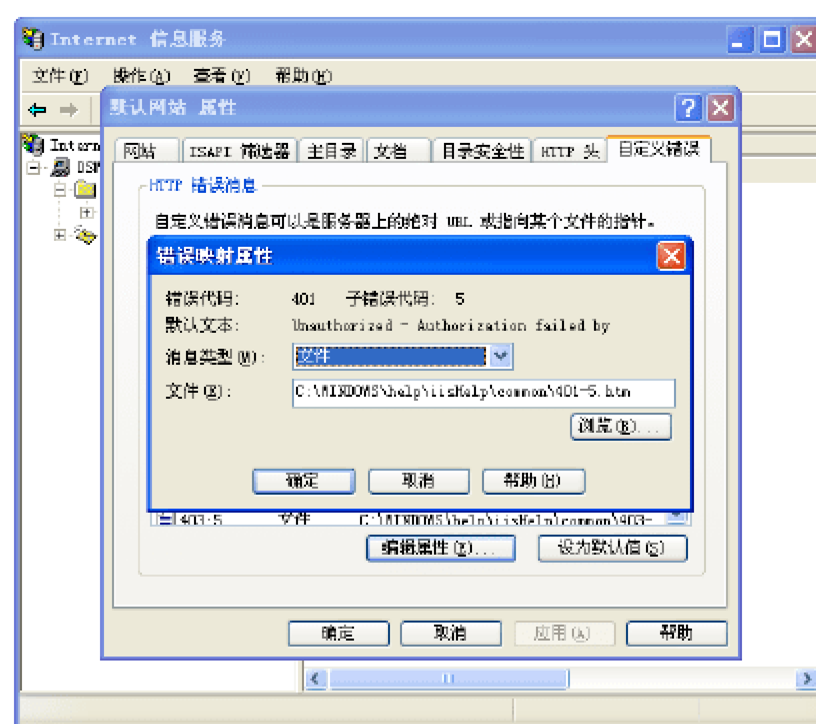


图 12-22　输入指向自定义错误消息的 URL

(5) 单击“确定”按钮，即可完成自定义错误的添加。

如果定义了用于处理自定义错误的 ASP 文件，则脚本将作为传送上下文的 URL 被调用。例如，如果有一个定义好的用于处理 HTTP 405 错误的 ASP 文件 Handle405.asp，则当用户打开网页时出现 405 错误，就会调用此 ASP 文件。

如果使用静态自定义错误文件(HTML 文件)，则应将消息类型设为“文件”。如果打算开发一个应用程序(使用 ISAPI 或 ASP)来处理错误，则应将消息类型设为 URL。应该注意，错误状态是用 URL 参数传递给应用程序的，而设置 HTTP 头状态则是应用程序的职责；否则，HTTP 的反应状态就是 HTTP1.1200OK。

12.3.2　自定义 ASP 错误处理

Web 站点及其所有应用程序都默认将 ASP 错误进程传送到 500-100.asp 文件，该文件处理 ASP 文件编译和运行期间发生的任何错误。当 ASP 文件发生错误时，IIS 通过 Server 对象的 Transfer 方法将请求从正在执行的 ASP 文件传输到 500-100.asp 文件，原来请求的 ASP 文件立即终止并返回详细的错误信息，如发生错误的行号和对错误的描述等。

然而，对于用户创建的任何新的 Web 站点或其应用程序来讲，500-100.asp 错误进程默认不是传送到自定义错误处理文件，而是传送到 HTTP 1.1 默认错误处理。可以在 IIS 管理单元中将 500-100.asp 错误映射到 500-100.asp 文件，或者映射到自己创建的 ASP 错误处理文件。

要创建自己的 ASP 错误处理文件，需要使用 Server 对象的 GetLastError 方法，该方法返回一个 ASPError 对象。ASPError 对象用来捕捉 ASP 错误并向用户返回有用的信息，如错误描述和发生错误的行号等。ASPError 对象包含如表 12-6 所示的只读属性。

表 12-6　ASPError 对象的属性

属　性	说　明
ASPCode	返回 ASP 产生的错误码
Number	返回标准 ASP 错误码
Source	返回产生错误行的实际源代码
Category	返回错误类型
File	错误发生时，返回正被处理的 ASP 文件
Line	返回发生错误的行
Column	返回发生错误的列
Description	返回错误的简短描述
ASPDescription	如果是 ASP 相关错误，则返回更详细的错误描述

【练习 12-5】自定义一个 ASP 错误处理。

(1) 创建一个名为 500-100ac.asp 的文件，其代码如下所示：

```
<%
Set objAE=Server.GEtLastError
Response.Write objAE.Category&"<Br>"
If objAE.ASPCode>"" Then
    Response.Write "错误码："&objAE.ASPCode&"<Br>"
End If
Response.Write Hex(objAE.Number)&"<Br>"
Response.Write "<B>"&objAE.Description&"</B><Br>"
If objAE.Description>"" Then
Response.Write objAE.Description&"<Br>"
End If
blnErrorWritten=False
If objAE.Source>"" Then
    strServerName=LCase(Reuqest.ServerVariables("SERVER-NAME"))
    strServerIP=Request.ServerVariables("LOCAL_ADDR")
    strRemoteIP=Reuqest.ServerVariables("REMOTE_ADDR")
    If (strServerName="localhost" Or strServerIP=strRemoteIP)_
    And objAE.File<>"?" Then
    Response.Write objAE.File
    If objAE.Line>0 Then
      Response.Write ",错误所在行"&objAE.Line
    End If
    If objAE.Column>0 Then
      Response.Write ",错误所在列"&objAE.Column&"<Br>"
    End If
    Response.Write Server.HTMLEncode(objAE.Source)&"<Br>"
    blnErrorWritten=True
    End If
End If
If Not blnErrorWritten And objAE.File<>"?" Then
Response.Write "<B>"&objAE.File
If objAE.Line>0 Then
Response.Write "，错误所在行"&objAE.Line
End If
If objAE.Column>0 Then
Response.Write "，错误所在列"&objAE.Column&"</B><Br>"
End If
End If
%>
```

(2) 将 500-100ac.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 在“Internet 信息服务”控制台中右击待映射的 Web 站点、虚拟目录或目录，然后在弹出的快捷菜单中选择“属性”命令，打开相应的属性对话框。

(4) 切换到“自定义错误”选项卡，选择 500：100 错误。

(5) 单击“编辑属性”按钮，打开“错误映射属性”对话框。

(6) 在“消息类型”下拉列表框中选择 URL，在 URL 文本框中输入指向 500-100ac.asp 的 URL，如图 12-23 所示。

(7) 单击“确定”按钮。

(8) 将 500：100 错误映射为自定义错误处理文件后，若请求应用程序中的 ASP 文件时发生错误，就会显示自定义错误消息。例如，执行上面输入的脚本时，将显示如图 12-24 所示的错误提示页面。

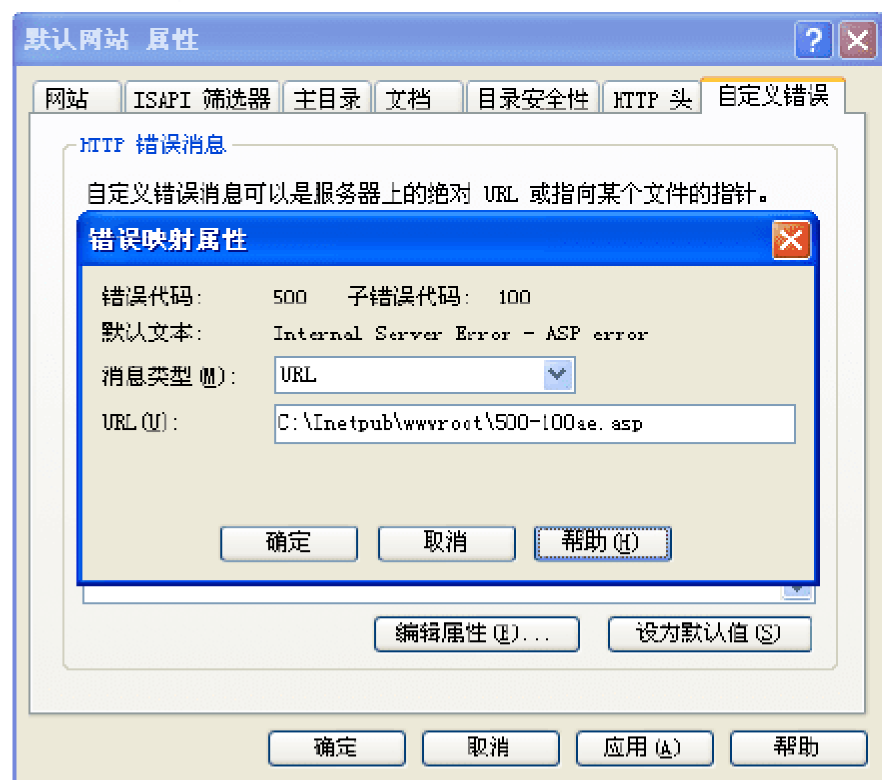


图 12-23　映射自定义 ASP 错误处理

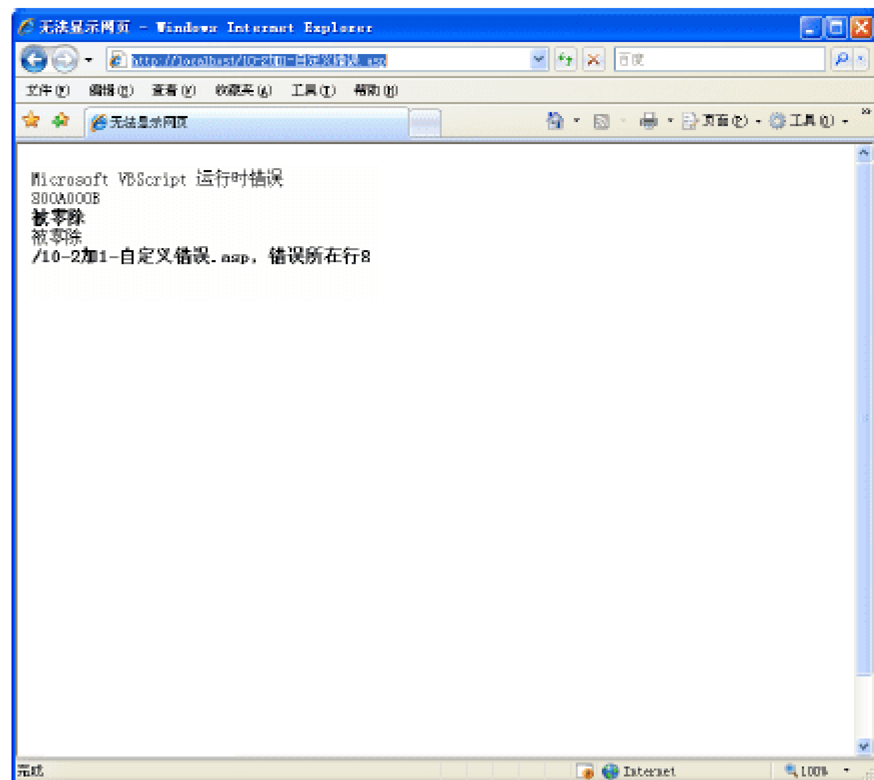


图 12-24　显示自定义错误信息

如果错误处理文件包含运行错误，则只显示此错误，而不考虑请求的.asp 文件中的错误类型。如果错误处理文件有预处理和编辑错误，则在浏览器中显示这两种错误，即显示错误处理文件中的错误和请求的 ASP 文件中的错误

12.3.3　错误跳转语句

当程序出现错误时将终止 ASP 的运行，此时可使用 On Error Resume Next 语句，使程序中的所有错误都被忽略，而自动执行下一条语句。这样程序就会完全执行，用户也不会看到出错信息。

在 ASP 中，可在程序最底端放上代码来处理错误，并在每个 ASP 程序中都使用缓冲区。如果错误发生，页面就会停止，页面内容也会被清除。

【练习 10-6】编写一段错误处理代码，在程序出错时能够清除页面中的显示内容并向用户返回错误的相关信息，程序运行效果如图 12-26 所示。

(1) 创建一个名为“10-3-错误调试.asp”的文件，其代码如下所示：

```
<%@ LANGUAGE="VBScript" %>
<%Response.Buffer = True
"设置 buffer 为 True
On Error Resume Next
"开始错误处理
```

```
%>
<%"错误处理
If Err.Number <> 0 Then
"清除页面
Response.Clear
"显示错误信息给用户
%>
<HTML>
<HEAD>
<TITLE></TITLE>
</HEAD>
<BODY BGCOLOR="#C0C0C0">
<FONT FACE='ARIAL'>程序运行时发生错误<BR>
请根据下面的错误提示信息检测排错。
<P><B>错误描述信息</B><BR>
错误号码：  <%= Err.Number %><BR>
错误信息：  <%= Err.Description %><BR>
出错文件：  <%= Err.Source %><BR>
出错行：  <%= Err.Line %><BR>
</FONT>
</BODY>
</HTML>
<%End If%>
```

(2) 将上面输入的代码保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 在浏览器中打开网页后，效果如图 12-25 所示。

(4) 在页面的文本框中输入数字－2 后，单击“提交”按钮，网页将显示如图 12-26 所示的效果。

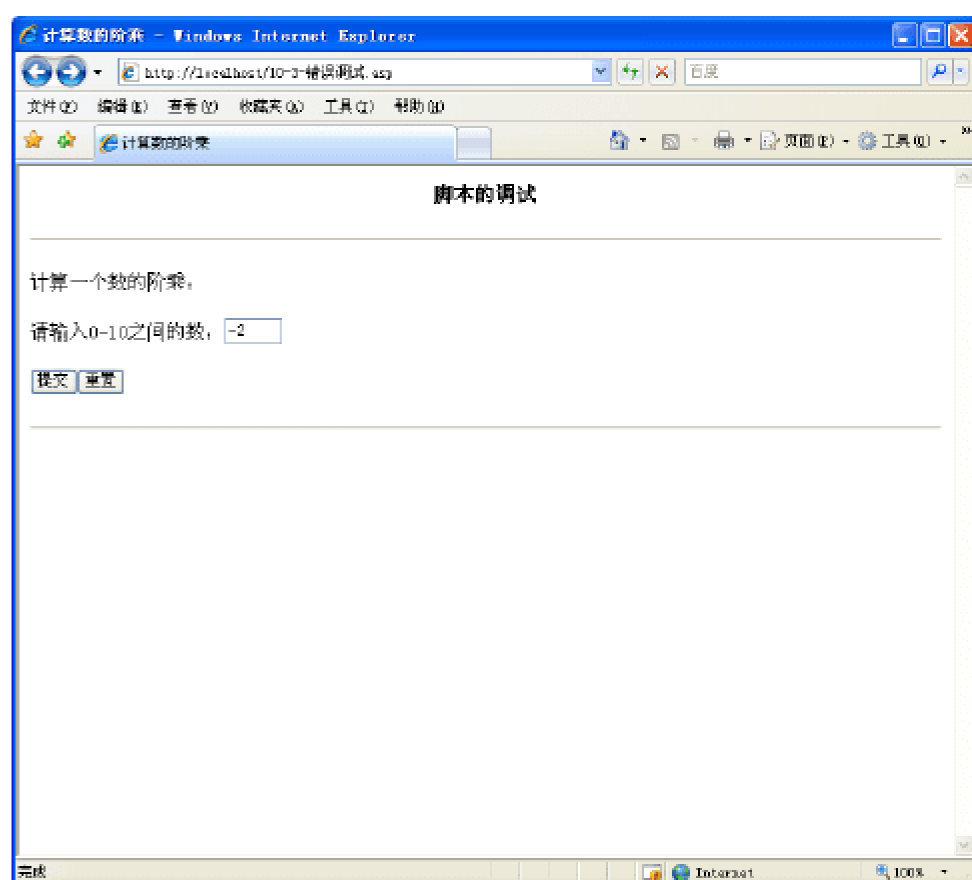


图 12-25　脚本调试页面

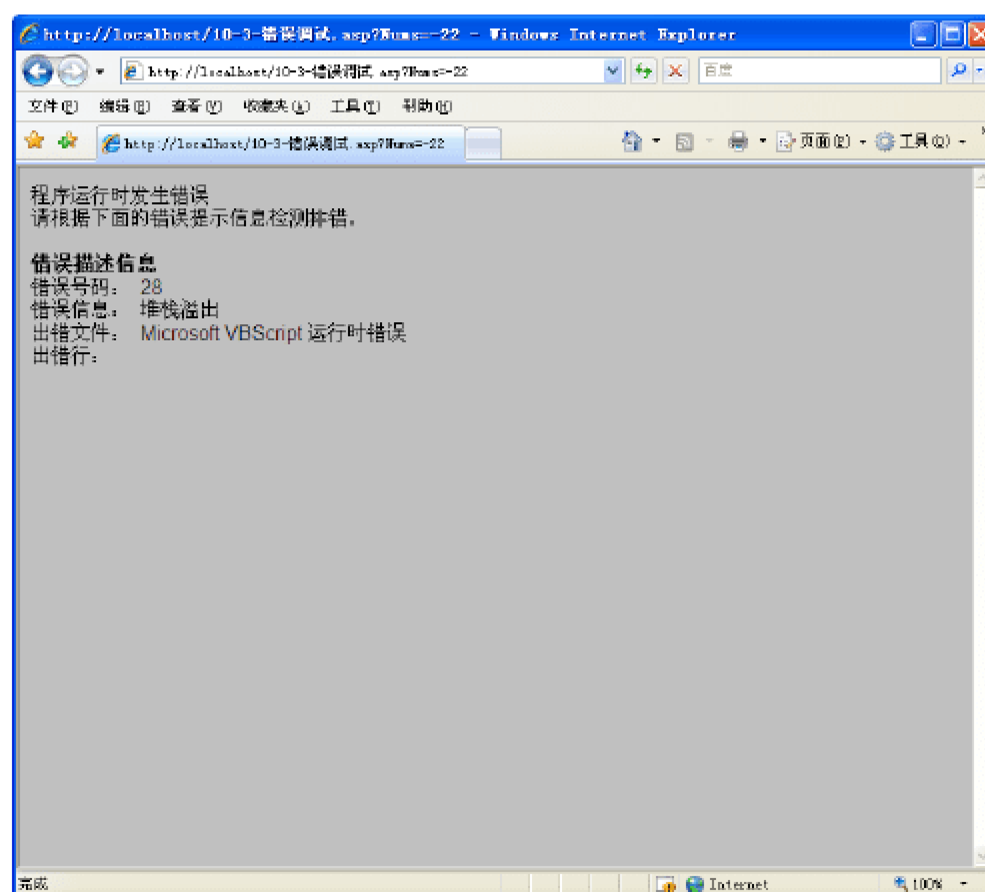


图 12-26　错误处理页面

注意：

在程序的开始部分添加一条 On Error Resume Next 语句，这样当运行出现错误时也不会影响程序的执行。

在数据库的执行中不能直接加入 On Error Resume Next 错误处理语句，因为如果在数据库添加记录过程中发生运行错误，On Error Resume Next 语句会忽略所有错误，程序依旧会向数据库中添加一个错误的信息数据。为避免这种情况，可在对数据库进行数据处理前添加判断语句，如下例所示：

```
If Err.Number = 0 And objConnection.Errors.Count = 0 Then
"这里才能执行语句，因为没有错误
Set rstResults = dbData.Execute(txtSql)
End If
```

12.4　事　　务

事务是一种服务器操作，该操作要么全部成功，要么全部失败，即使该操作包含许多步骤(如订购、检查存货及付账等)也是如此。可以创建在事务内运行的服务器端脚本，如果脚本出错，则整个事务将中止。

ASP 事务基于“组件服务(Component Services)”事务环境，这是一种事务处理系统，用来开发、部署和管理高性能、可扩展的 Internet/Intranet 服务器应用程序。如果事务中止，“组件服务”将恢复对支持事务的资源所做的任何更改。

12.4.1　事务简介

ASP 事务处理是以 Microsoft Transaction Server(MTS)为基础的。Transaction Server 为开发分布式的、基于组件的应用程序提供了一个应用程序设计模型，也为配置和管理这些应用程序提供了一个运行环境。

事务是整体成功或失败的操作。事务处理用于对数据库进行可靠的更新。在对数据库进行许多相关更改或同时更新多个数据库时，要保证所有更改都被正确执行。如果这些更改中的任何一个失败，都须要恢复数据库表的原始状态。

事务不能跨越多个 ASP 页。如果一个事务需要来自多个组件的对象，则将须使用这些对象的操作组合在一个 ASP 页中。例如，假定有一个组件用于更新工资单数据库，还有一个组件用于更新人力资源数据库中的员工记录。为了记录一个员工的新工资信息，则须要编写在一个事务环境中调用这两个组件的脚本，一个用于更新工资单数据库，另一个用于更新人力资源数据库中的员工等级。

注意：

事务处理只适用于数据库访问，MTS 不能对文件系统或其他的非事务性资源的更改进行恢复操作。应用程序所访问的数据库必须为 MTS 所支持。

12.4.2　声明事务脚本

将一个页声明为事务性时，此页中的任何脚本命令和对象都运行在同一个事务环境中。Transaction Server 处理生成事务的细节并决定事务成功(提交)或失败(终止)。要声明事务脚本，只需在 ASP 页顶部添加@TRANSACTION 命令即可，其语法结构如下：

```
<%@ TRANSACTION = value %>
```

value 参数可以是下列常量之一。

- Requires_New：启动一个新的事务。
- Required：启动一个新的事务。
- Supported：不启动事务。
- Not_Supported：不启动事务。

大多数应用程序只有一些特定的操作需要事务环境。如航空公司站点可以用事务脚本来售票和分配座位等。由于事务只适用于需要事务进程的页，因此不可将应用程序的 Global.asa 文件声明为事务性。

通过调用 Server 对象的 Transfer 和 Execute 方法，事务可以扩展到多个 ASP 页。如果脚本包含值为 Required 的@TRANSACTION 命令，并且 Server.Transfer 或 Server.Execute 方法调用该脚本，则脚本将继续运行调用 ASP 文件的事务(若调用 ASP 文件已被处理)。如果调用 ASP 文件未被处理，则被调用的 ASP 文件将自动创建新事务。例如：

ASP1:

```
<%@TRANSACTION =Required%>
<%
……
Server.Transfer("/asp/asp2.asp")
%>
```

ASP2:

```
<%@TRANSACTION =Required%>
<%
Set objSales=Server.CreateObject("SalesTransacted.Complete")
……
%>
```

两个脚本的交互只能形成一个单一事务。

注意：

@TRANSACTION 指令必须在一页中的第一行，否则将产生错误。必须将该指令添加到需要在事务下运行的每一页中。当脚本处理结束时，当前事务即告结束。

12.4.3　提交或中止事务

因为 Transaction Server 跟踪事务处理，所以它决定事务是完全成功还是失败。可以用 ObjectContext 对象来提交和中止由“组件服务”管理的事务。ObjectContext 对象包含 SetAbort 和 SetComplete 两个方法。

SetAbort 方法用来中止事务。例如，当一个事务从一个组件收到错误消息、违反商业规范时(例如，账户余额小于 0)或读写文件等非事务性操作失败时，脚本就须要终止该事务。如果网页在事务完成之前超时，则事务也将被中止。

调用 SetAbort 方法，将导致“组件事务”恢复对资源所做的任何更改。事务中止时，脚本的 OnTransactionCommit 事件将被处理。如下例所示：

```
<%@ TRANSACTION=Required LANGUAGE=VBScript%>
<%
ObjectContext.SetAbort
Sub OnTransactionAbort
Response.Write "<P><B>事务被中止</B>"
End Sub
%>
```

调用 SetComplete 方法并不意味着事务被提交。仅当脚本调用的所有事务性组件都调用 SetComplete 方法时，事务才提交。大多数情况下，不需要在脚本内部调用 SetComplete 方法，因为只要脚本没有调用 SetAbort 方法，则脚本处理完后将提交当前事务。提交事务时将触发 OnTransactionAbort 事件。如下例所示：

```
<%@ TRANSACTION=Required LANGUAGE=VBScript%>
<%
ObjectContext.SetComplete
Sub OnTransactionCommit
Response.Write "<P><B>事务被提交</B>"
End Sub
%>
```

12.5　习　　题

12.5.1　填空题

(1) ASP 程序在运行时有可能出现各种错误，这些错误可分为 3 类：____________、____________和____________。

(2) ____________是在脚本中作了标记的位置，在指定的脚本行暂停执行脚本。

(3) 为了能够在第一次调用脚本时暂时中断执行，可以在脚本的可疑行之前插入一条____________语句。执行到该语句时，脚本会暂停执行。

(4) 自定义错误消息可为两种形式：映射为____________或映射为____________。

(5) 使用____________语句，所有的错误都会被忽略，程序会自动执行下一条语句。

(6) 在 ASP 页顶部添加____________命令可声明事务脚本。

12.5.2　选择题

(1) 下面不属于 Microsoft 脚本编辑器的主要功能有(　　)。

A. 编辑和修改服务器端脚本

B. 在服务器端脚本执行时，打开命令窗口监视变量、属性或数组元素的值

C. 在脚本的指定行设置断点以暂停执行服务器端脚本

D. 运行服务器端脚本时跟踪过程/函数

(2) 下面关于事务处理说法不正确的是(　　)。

A. 事务是整体成功或失败的操作

B. 大多数应用程序只有一些特定的操作需要事务环境

C. Transaction Server 决定事务是完全成功还是失败

D. 事务可以跨越多个 ASP 页

(3) 下面关于事务提交说法正确的是(　　)。

A. 调用 SetComplete 方法意味着事务被提交

B. 仅当脚本调用的所有事务性组件都调用 SetComplete 方法时，事务才提交

C. 大多数情况下脚本内部不可缺少对 SetComplete 方法的调用

D. 提交事务时将触发 OnAbort 事件

12.5.3　问答题

(1) 什么是断点调试，它有什么作用？

(2) 在数据库的执行中如何加入 On Error Resume Next 错误处理语句？

第13章　综 合 实 例

本章将通过几个简单的实例，综合本书前面各个章节所学习的内容，详细介绍利用 ASP 结合 Access 数据库设计动态 ASP 网站模块的具体方法。通过本章的理论学习和练习，用户应了解和掌握以下内容：

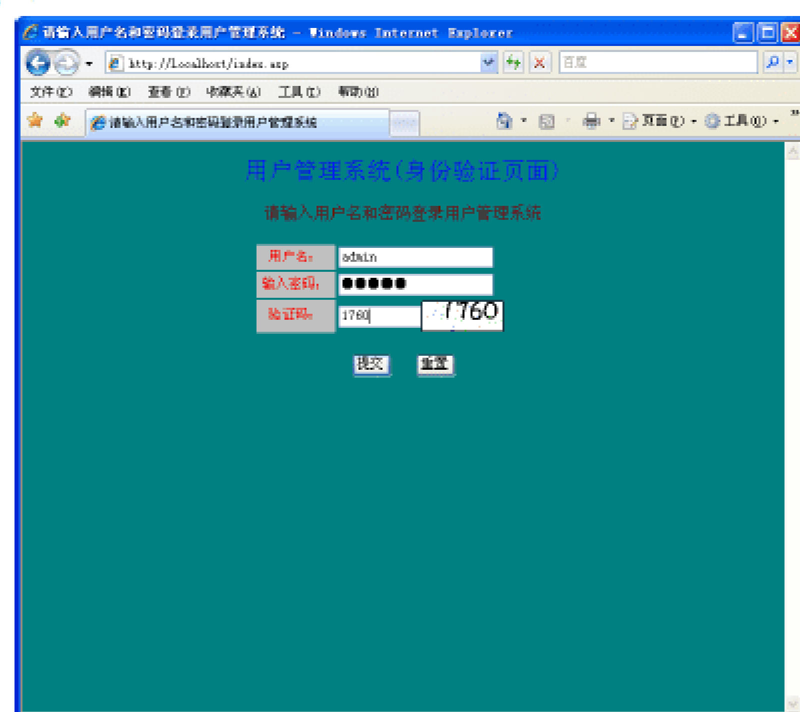
- 掌握用户管理的创建方法
- 掌握网站留言板的创建方法
- 掌握网络聊天室的创建方法
- 掌握网站流量统计的创建方法

13.1　用 户 管 理

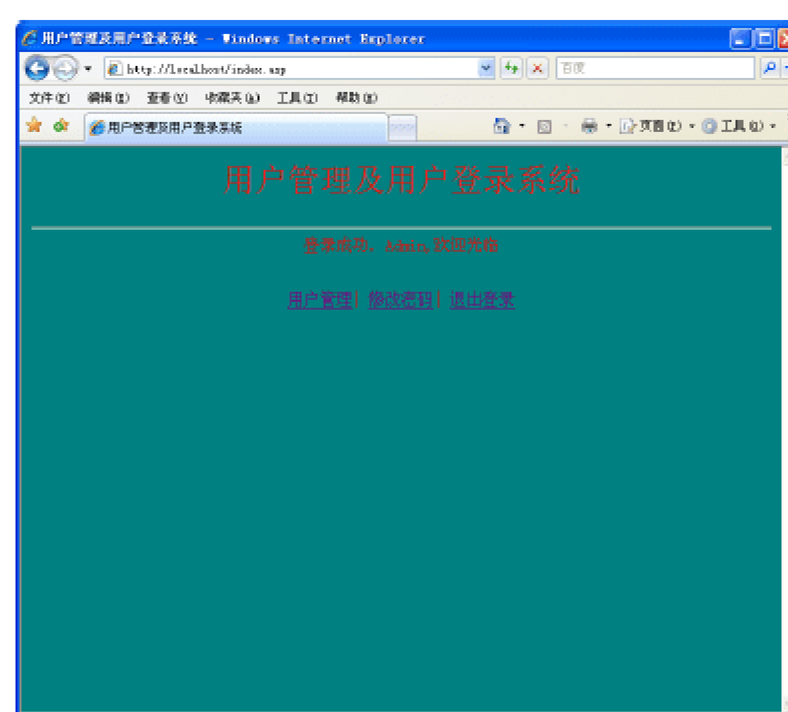
用户管理是目前网站中最常见的模块，它可以帮助站点收集和分析众多的网络客户信息。例如通过获取的用户注册资料对访问网站的人群进行统计和归类，并以此为依据为网站的发展提供合理的规划。

13.1.1　功能介绍

本章所介绍的用户管理系统的主界面如图 13-1 左图所示。用户可以在该界面中输入登录信息登录用户管理界面，ASP 页面将会根据用户的登录身份，打开相应的用户管理页面。在系统管理员页面中，用户可以根据需要执行“用户管理”、“修改密码”和“退出登录”命令(如图 13-1 右图所示)，而在普通用户页面中，用户只能执行“密码修改”和“退出登录”命令。



用户管理的主界面



用户管理界面(管理员)

图 13-1　用户管理模块

在系统管理员页面中，用户可以单击“用户管理”链接打开用户列表页面，在该页面中系统管理员不仅可以查看所有的用户和增加新用户，还能够对特定的用户进行修改和删除操作。

13.1.2　模块设计

网站的用户管理模块需要具备安全性、有效性，合法性等多方面的性能，主要包括如图 13-2 所示的用户登录页面、系统管理界面 、普通用户管理界面、管理员退出登录、密码修改页面、用户管理界面、添加普通用户、修改普通用户、删除普通用户、修改用户密码页面和普通用户退出登录等模块。

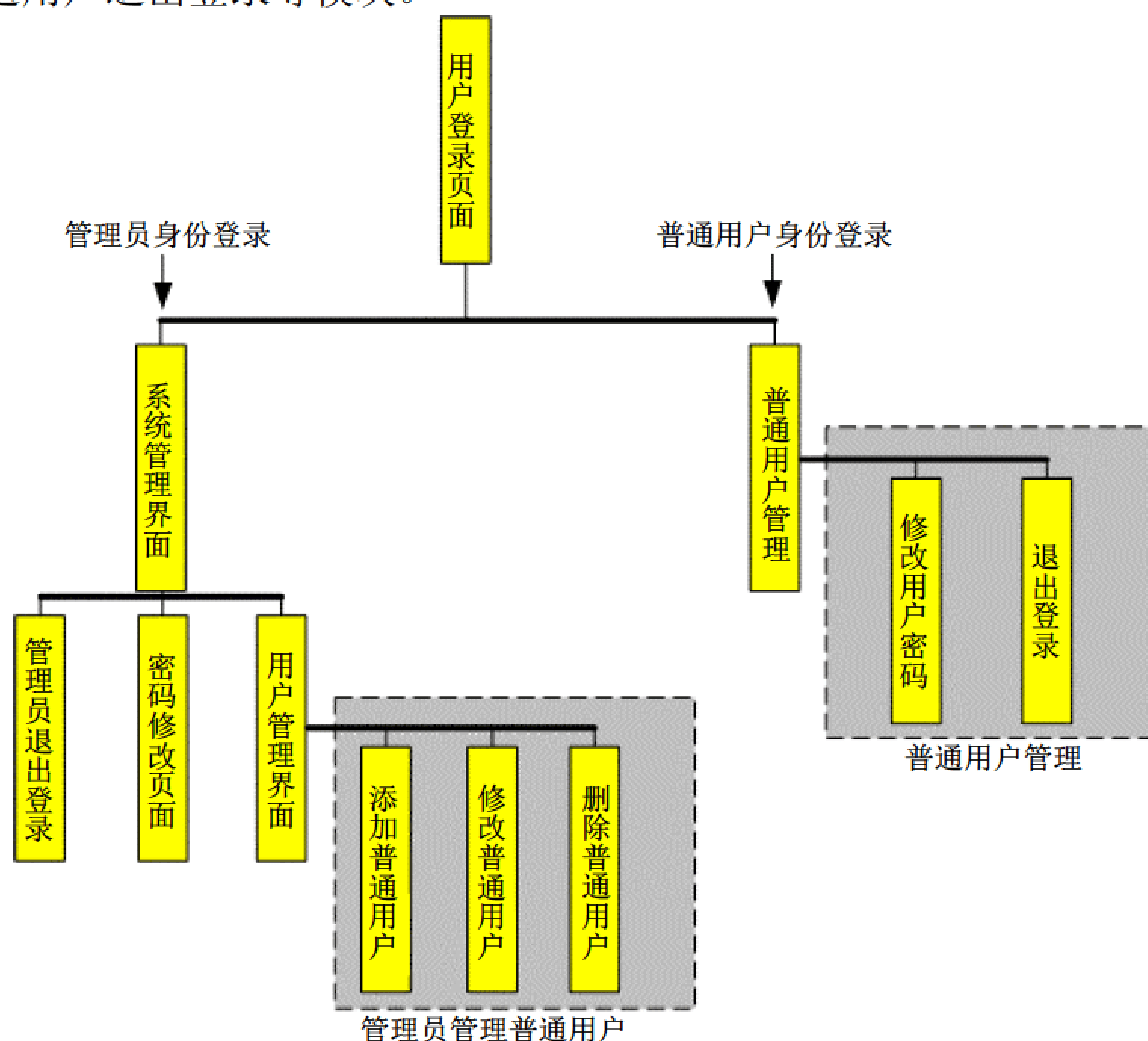


图 13-2　用户管理模块的结构

1. 页面文件设计

从用户管理模块的结构可以看出，该模块主要由以下几个页面文件组成。

- Chkwd.asp 文件：该文件要求用户输入用户名、密码和验证码，登录用户管理系统。
- Index.asp 页面：该页面是用户管理系统的主页面(如图 13-3 所示)，可以根据登录用户的权限显示不同的操作链接，如图 13-4 所示。
- UserList.asp 页面：所有用户名称的显示页面，如图 13-5 所示，系统管理员可以在该页面中修改或删除系统中的普通用户。

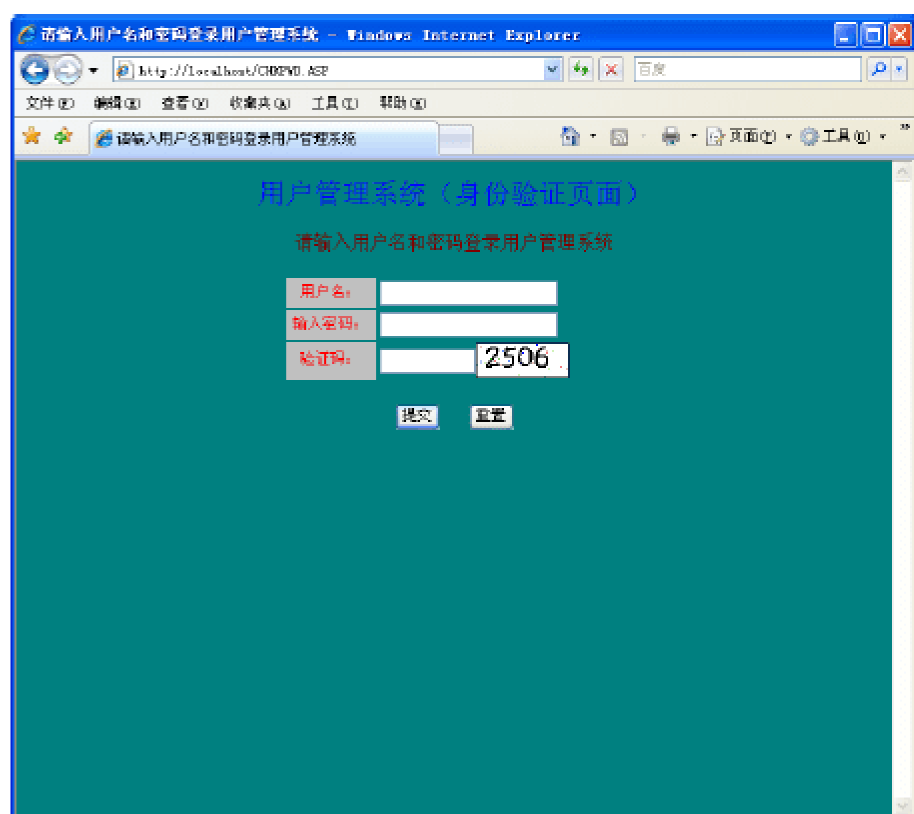


图 13-3　Index.asp 页面

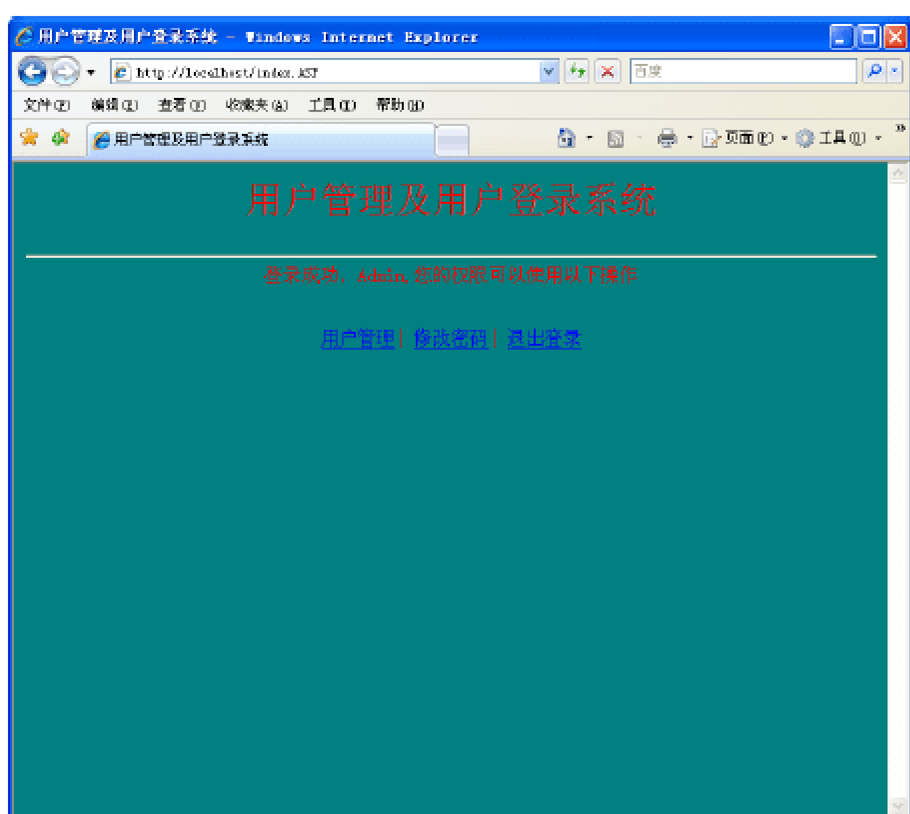


图 13-4　用户登录后的页面

- UserAdd.asp 页面：在该页面中，系统管理员可以添加新的普通用户，如图 13-6 所示。



图 13-5　UserList.asp 页面

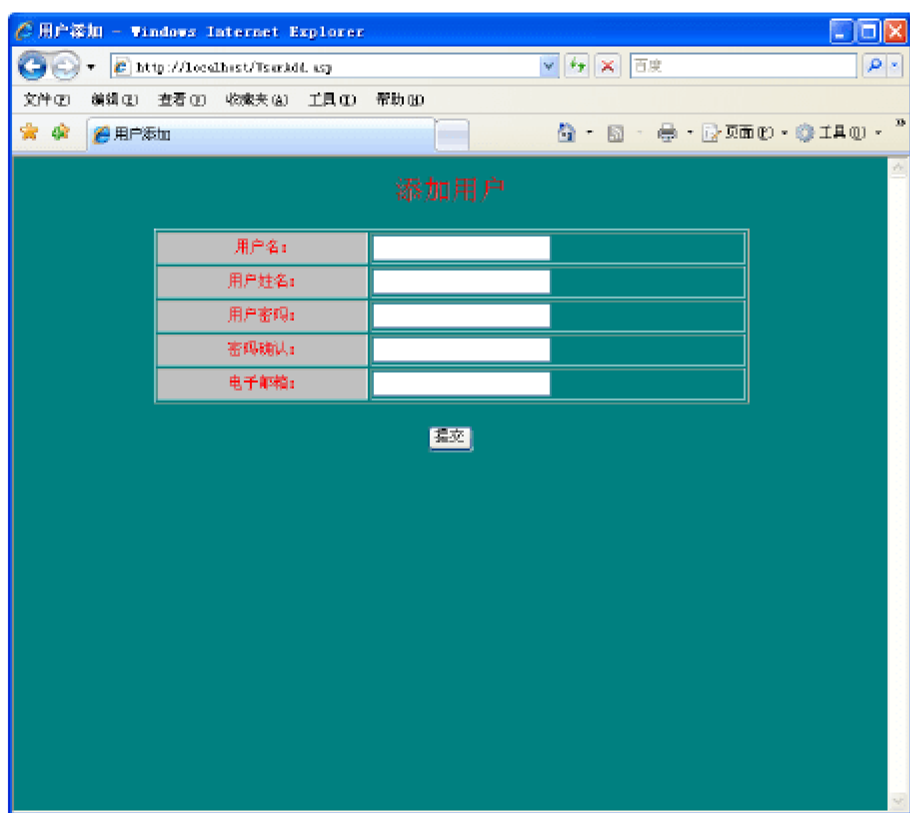


图 13-6　UserAdd.asp 页面

- UserEdit.asp 页面：在该页面中，系统管理员可以修改普通用户的基本信息，如图 13-7 所示。
- UserSave.asp 文件：该文件完成用户添加和用户信息修改时的数据保存工作。
- UserDel.asp 页面：该页面完成用户的删除。
- PwdChange.asp 页面：该页面可以完成用户密码的修改，如图 13-8 所示。
- SavePwd.asp 页面：该页面完成用户新密码在数据库中的保存。
- Logout.asp 页面：该页面是用户退出登录页面。
- Conn.asp 文件(公用文件)：该文件是数据库连接文件。
- Md5.asp 文件(公用文件)：该文件完成用户密码的加密。
- procedure.asp 文件(公用文件)：该文件中包含邮件地址格式确认和用户名是否在数据库中已存在这两个过程。

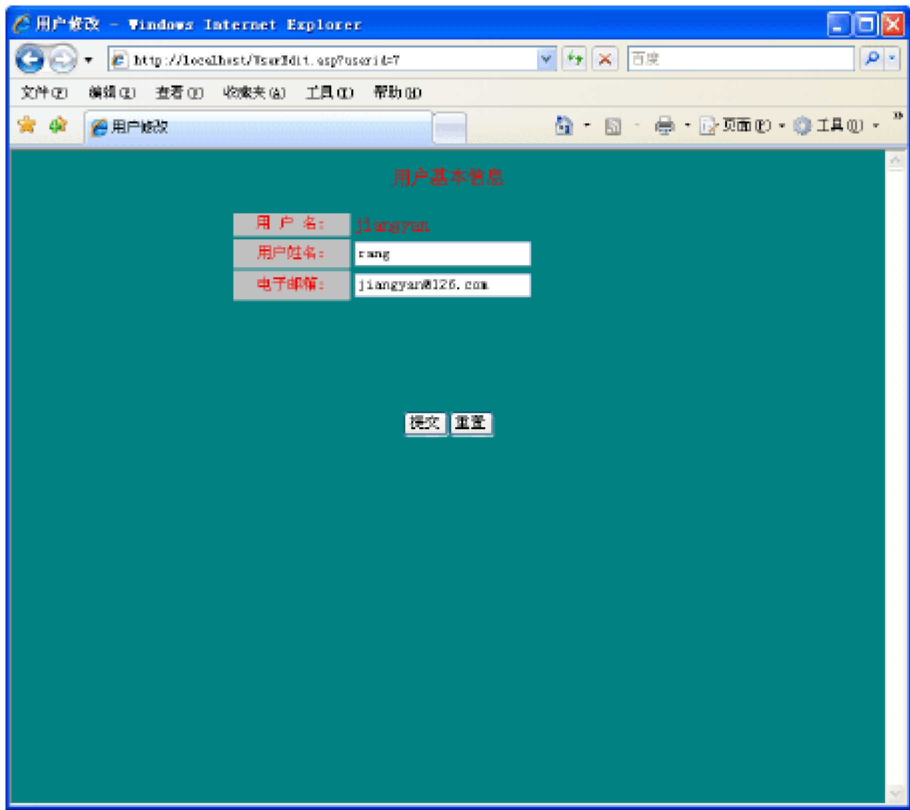


图 13-7　UserEdit.asp 页面

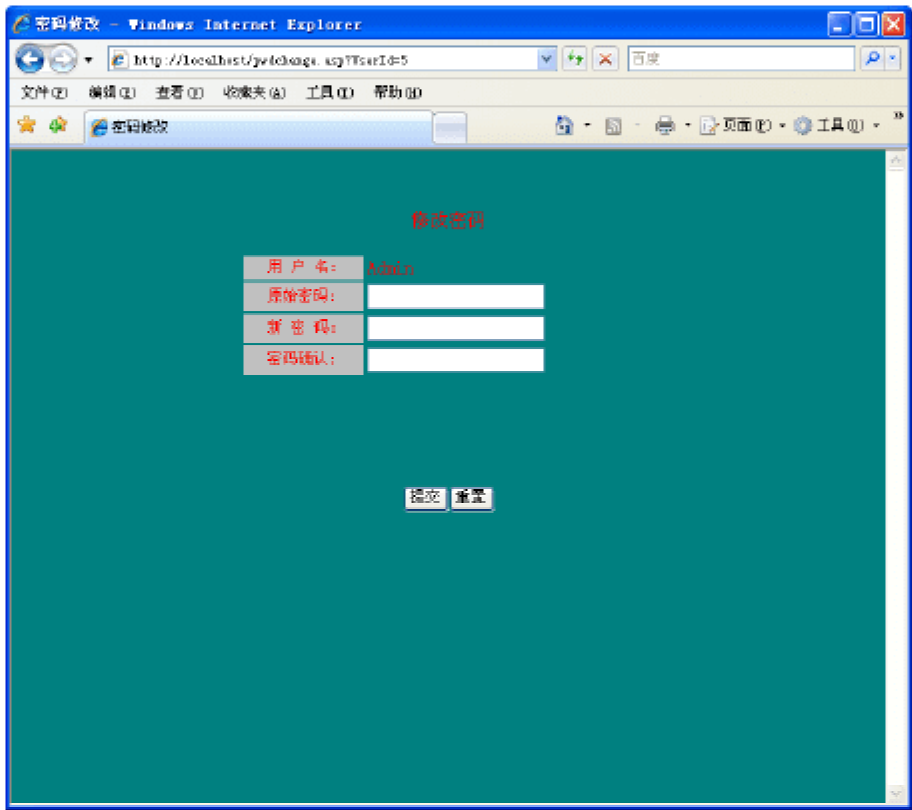


图 13-8　PwdChange.asp 页面

注意：

在用户管理系统中，除了以上所介绍的几个主要文件以外，还有 Function.asp、Safecode.asp、Canvas.asp 和 Font.asp 等页面文件。这些文件完成用户登录时验证码的生成与显示。

2. 数据库设计

本章实例采用 Access 数据库。由模块结构可以看出，用户管理模块中只需要使用一个 Users 数据表即可，用来存储用户的基本信息，例如登录的普通用户名、密码、用户姓名和用户资料(电子邮箱地址)等，如表 13-1 所示。

表 13-1　Users 数据表的结构

字　　段	字 段 类 型	主 关 键 字	说　　明
User_ID	自动编号	*	编号
User_Name	文本	—	用户登录名
User_Pwd	文本	—	用户登录密码
Name	文本	—	用户姓名
Email	文本	—	电子邮箱

13.1.3　实例操作

本节将根据本章第 13.1.2 所介绍的系统结构，通过实例介绍创建一个用户管理系统的方法。用户在进行实例操作之前，应结合本书第 1.2 节和第 1.3 节所介绍的方法在本地计算机上构建一个 ASP 应用程序的开发环境。

1. 创建数据库

下面将以表 13-1 所示的数据库结构为基础，为用户管理系统创建一个数据库 UserMan。

【练习 13-1】在 C:\Inetpub\wwwroot 文件夹中创建一个名为 UserMan 的数据库文件，然后按照表 13-1 所示的内容在数据库中创建数据表 Users。

(1) 选择“开始”|“程序”| Microsoft Office| Access 2003 命令，启动 Access 2003 数据库，然后选择“文件”|“新建”命令，打开“新建文件”对话框。

(2) 单击“空数据库”按钮，在打开的“文件新建数据库”对话框的“文件名”文本框中输入 UserMan。

(3) 单击“创建”按钮，在打开的 Access 2003 的工作界面的工作区域中选择“对象”列表框中的“表”选项，然后在对话框右侧的选项区域中双击“使用设计器创建表”选项打开数据表“表”的设计检视窗口。

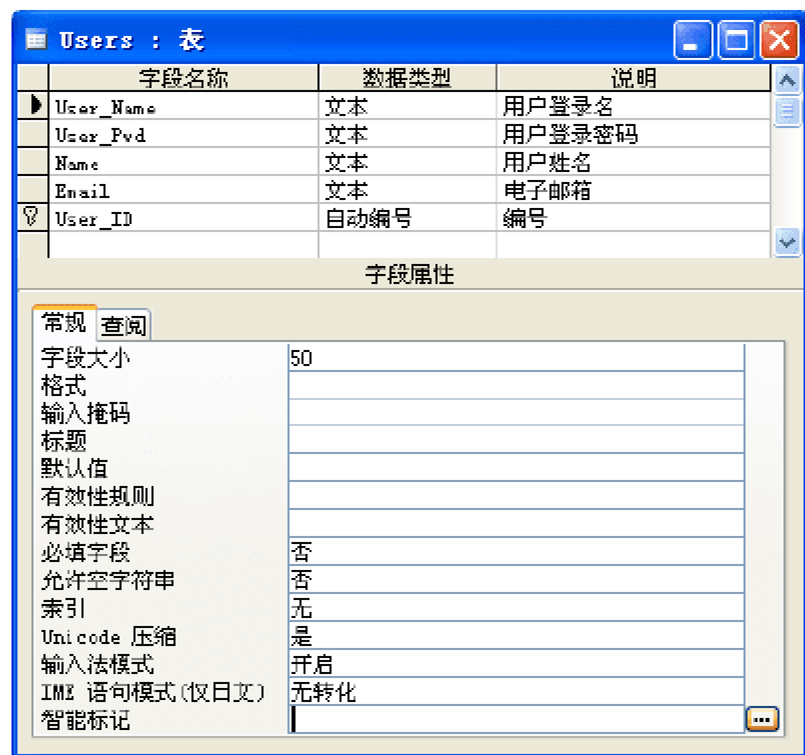
(4) 设置数据表中字段的结构。在“表”的设计检视窗口的“字段名称”列的第一个单元格中输入 User_ID，在其后的“数据类型”下拉列表中选择“自动编号”选项，在“说明”列的第一个单元格中输入对表格字段的描述文本。

(5) 根据表 13-1 所示的内容，参考步骤(4)的操作，在“表”的设计检视窗口中设置 User_Name、User_Pwd、Name 和 Email 字段。

(6) 右击 User_ID 字段，在弹出的快捷菜单中选择“主键”命令，为 User_ID 字段前添加🔑标志，将该字段设置为主键。

(7) 完成以上操作后选择“文件”|“保存”命令，在打开的“另存为”对话框的“将表另存为”文本框中输入 Users，单击“确定”按钮保存数据表，完成后的效果如图 13-9 所示。

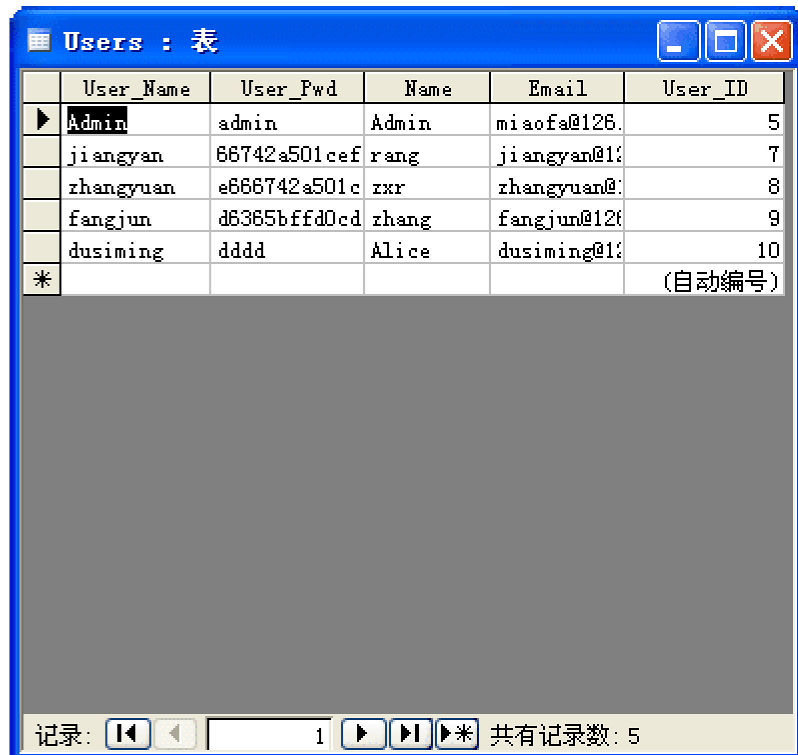
(8) 在 Access 的工作区域中双击 Users 表将其打开，并在数据表中输入如图 13-10 所示的数据。



字段名称	数据类型	说明
User_Name	文本	用户登录名
User_Pwd	文本	用户登录密码
Name	文本	用户姓名
Email	文本	电子邮箱
User_ID	自动编号	编号

字段属性	值
字段大小	50
格式	
输入掩码	
标题	
默认值	
有效性规则	
有效性文本	
必填字段	否
允许空字符串	否
索引	无
Unicode 压缩	是
输入法模式	开启
IME 语句模式(仅日文)	无转化
智能标记	

图 13-9　表 Users



User_Name	User_Pwd	Name	Email	User_ID
Admin	admin	Admin	miaofa@126.	5
jiangyan	66742a501cef	rang	jiangyan@1	7
zhangyuan	e666742a501c	zxr	zhangyuan@	8
fangjun	d6365bffd0cd	zhang	fangjun@12	9
dusiming	dddd	Alice	dusiming@1	10
				(自动编号)

图 13-10　PwdChange.asp 页面

(9) 选择“文件”|“保存”命令，保存数据表 Users，然后关闭 Access 数据表。

2. 创建数据库连接文件

完成数据库的创建工作后，用户可以参考下面练习所介绍的方法，创建 ASP 网页数据

库连接文件 Conn.asp。

【练习 13-2】创建数据库连接文件 Conn.asp，连接数据库 UserMan。

(1) 创建数据库连接文件 Conn.asp，其代码如下所示：

```
<%
'Dim databasename,conn,ConStr
databasename="UserMan.mdb"
ConStr="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.MapPath(databasename)
Set conn=server.createobject("adodb.connection")
conn.open ConStr
%>
```

(2) 将 Conn.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

3. 设置密码加密文件

完成 Conn.asp 文件的创建工作后，用户可以参考下面实例的方法创建用户管理系统的密码加密文件 MP5.asp。该文件采用 MP5(Message Digest Algorithm 5)数据加密算法，对用户保存在网站数据库中的信息进行加密。

【练习 13-3】创建用户管理系统的密码加密文件 MP5.asp。

(1) 创建密码加密文件 MP5.asp，其代码如下所示：

```
<%
Private Const BITS_TO_A_BYTE = 8
Private Const BYTES_TO_A_WORD = 4
Private Const BITS_TO_A_WORD = 32
Private m_lOnBits(30)
Private m_l2Power(30)
Private Function LShIft(lValue, iShIftBits)
If iShIftBits = 0 Then
LShIft = lValue
Exit Function
ElseIf iShIftBits = 31 Then
If lValue And 1 Then
LShIft = &H80000000
Else
LShIft = 0
End If
Exit Function
ElseIf iShIftBits < 0 Or iShIftBits > 31 Then
Err.Raise 6
End If
If (lValue And m_l2Power(31-iShIftBits)) Then
```

```
LShIft = ((lValue And m_lOnBits(31-(iShIftBits + 1))) * m_l2Power(iShIftBits)) Or &H80000000
Else
LShIft = ((lValue And m_lOnBits(31-iShIftBits)) * m_l2Power(iShIftBits))
End If
End Function
Private Function RShIft(lValue, iShIftBits)
If iShIftBits = 0 Then
RShIft = lValue
Exit Function
ElseIf iShIftBits = 31 Then
If lValue And &H80000000 Then
RShIft = 1
Else
RShIft = 0
End If
Exit Function
ElseIf iShIftBits < 0 Or iShIftBits > 31 Then
Err.Raise 6
End If
RShIft = (lValue And &H7FFFFFFE) \ m_l2Power(iShIftBits)
If (lValue And &H80000000) Then
RShIft = (RShIft Or (&H40000000 \ m_l2Power(iShIftBits-1)))
End If
End Function
Private Function RotateLeft(lValue, iShIftBits)
RotateLeft = LShIft(lValue, iShIftBits) Or RShIft(lValue, (32-iShIftBits))
End Function
Private Function AddUnsigned(lX, lY)
    Dim lX4
    Dim lY4
    Dim lX8
    Dim lY8
    Dim lResult
    lX8 = lX And &H80000000
    lY8 = lY And &H80000000
    lX4 = lX And &H40000000
    lY4 = lY And &H40000000
lResult = (lX And &H3FFFFFFF) + (lY And &H3FFFFFFF)
If lX4 And lY4 Then
lResult = lResult Xor &H80000000 Xor lX8 Xor lY8
ElseIf lX4 Or lY4 Then
If lResult And &H40000000 Then
lResult = lResult Xor &HC0000000 Xor lX8 Xor lY8
```

```
Else
lResult = lResult Xor &H40000000 Xor lX8 Xor lY8
End If
Else
lResult = lResult Xor lX8 Xor lY8
End If
AddUnsigned = lResult
End Function
Private Function md5_F(x, y, z)
md5_F = (x And y) Or ((Not x) And z)
End Function
Private Function md5_G(x, y, z)
md5_G = (x And z) Or (y And (Not z))
End Function
Private Function md5_H(x, y, z)
md5_H = (x Xor y Xor z)
End Function
Private Function md5_I(x, y, z)
md5_I = (y Xor (x Or (Not z)))
End Function
Private Sub md5_FF(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(md5_F(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub
Private Sub md5_GG(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(md5_G(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub
Private Sub md5_HH(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(md5_H(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub
Private Sub md5_II(a, b, c, d, x, s, ac)
    a = AddUnsigned(a, AddUnsigned(AddUnsigned(md5_I(b, c, d), x), ac))
    a = RotateLeft(a, s)
    a = AddUnsigned(a, b)
End Sub
Private Function ConvertToWordArray(sMessage)
    Dim lMessageLength
    Dim lNumberOfWords
```

```
        Dim lWordArray()
        Dim lBytePosition
        Dim lByteCount
        Dim lWordCount
        Const MODULUS_BITS = 512
        Const CONGRUENT_BITS = 448
        lMessageLength = Len(sMessage)
         lNumberOfWords = (((lMessageLength + ((MODULUS_BITS-CONGRUENT_BITS) \
            BITS_TO_A_BYTE)) \ (MODULUS_BITS \ BITS_TO_A_BYTE)) + 1) *
            (MODULUS_BITS \ BITS_TO_A_WORD)
        ReDim lWordArray(lNumberOfWords-1)
        lBytePosition = 0
        lByteCount = 0
        Do Until lByteCount >= lMessageLength
lWordCount = lByteCount \ BYTES_TO_A_WORD
lBytePosition = (lByteCount Mod BYTES_TO_A_WORD) * BITS_TO_A_BYTE
lWordArray(lWordCount) = lWordArray(lWordCount) Or LShIft(Asc(Mid(sMessage, lByteCount +
        1, 1)), lBytePosition)
lByteCount = lByteCount + 1
        Loop
        lWordCount = lByteCount \ BYTES_TO_A_WORD
        lBytePosition = (lByteCount Mod BYTES_TO_A_WORD) * BITS_TO_A_BYTE
        lWordArray(lWordCount) = lWordArray(lWordCount) Or LShIft(&H80, lBytePosition)
        lWordArray(lNumberOfWords-2) = LShIft(lMessageLength, 3)
        lWordArray(lNumberOfWords-1) = RShIft(lMessageLength, 29)
        ConvertToWordArray = lWordArray
End Function
Private Function WordToHex(lValue)
        Dim lByte
        Dim lCount
        For lCount = 0 To 3
lByte = RShIft(lValue, lCount * BITS_TO_A_BYTE) And m_lOnBits(BITS_TO_A_BYTE-1)
WordToHex = WordToHex & Right("0" & Hex(lByte), 2)
        Next
End Function
Public Function MD5(sMessage)
        m_lOnBits(0) = CLng(1)
        m_lOnBits(1) = CLng(3)
        m_lOnBits(2) = CLng(7)
        m_lOnBits(3) = CLng(15)
        m_lOnBits(4) = CLng(31)
        m_lOnBits(5) = CLng(63)
        m_lOnBits(6) = CLng(127)
```

```
m_lOnBits(7) = CLng(255)
m_lOnBits(8) = CLng(511)
m_lOnBits(9) = CLng(1023)
m_lOnBits(10) = CLng(2047)
m_lOnBits(11) = CLng(4095)
m_lOnBits(12) = CLng(8191)
m_lOnBits(13) = CLng(16383)
m_lOnBits(14) = CLng(32767)
m_lOnBits(15) = CLng(65535)
m_lOnBits(16) = CLng(131071)
m_lOnBits(17) = CLng(262143)
m_lOnBits(18) = CLng(524287)
m_lOnBits(19) = CLng(1048575)
m_lOnBits(20) = CLng(2097151)
m_lOnBits(21) = CLng(4194303)
m_lOnBits(22) = CLng(8388607)
m_lOnBits(23) = CLng(16777215)
m_lOnBits(24) = CLng(33554431)
m_lOnBits(25) = CLng(67108863)
m_lOnBits(26) = CLng(134217727)
m_lOnBits(27) = CLng(268435455)
m_lOnBits(28) = CLng(536870911)
m_lOnBits(29) = CLng(1073741823)
m_lOnBits(30) = CLng(2147483647)

m_l2Power(0) = CLng(1)
m_l2Power(1) = CLng(2)
m_l2Power(2) = CLng(4)
m_l2Power(3) = CLng(8)
m_l2Power(4) = CLng(16)
m_l2Power(5) = CLng(32)
m_l2Power(6) = CLng(64)
m_l2Power(7) = CLng(128)
m_l2Power(8) = CLng(256)
m_l2Power(9) = CLng(512)
m_l2Power(10) = CLng(1024)
m_l2Power(11) = CLng(2048)
m_l2Power(12) = CLng(4096)
m_l2Power(13) = CLng(8192)
m_l2Power(14) = CLng(16384)
m_l2Power(15) = CLng(32768)
m_l2Power(16) = CLng(65536)
m_l2Power(17) = CLng(131072)
```

```
m_l2Power(18) = CLng(262144)
m_l2Power(19) = CLng(524288)
m_l2Power(20) = CLng(1048576)
m_l2Power(21) = CLng(2097152)
m_l2Power(22) = CLng(4194304)
m_l2Power(23) = CLng(8388608)
m_l2Power(24) = CLng(16777216)
m_l2Power(25) = CLng(33554432)
m_l2Power(26) = CLng(67108864)
m_l2Power(27) = CLng(134217728)
m_l2Power(28) = CLng(268435456)
m_l2Power(29) = CLng(536870912)
m_l2Power(30) = CLng(1073741824)

Dim x
Dim k
Dim AA
Dim BB
Dim CC
Dim DD
Dim a
Dim b
Dim c
Dim d

Const S11 = 7
Const S12 = 12
Const S13 = 17
Const S14 = 22
Const S21 = 5
Const S22 = 9
Const S23 = 14
Const S24 = 20
Const S31 = 4
Const S32 = 11
Const S33 = 16
Const S34 = 23
Const S41 = 6
Const S42 = 10
Const S43 = 15
Const S44 = 21
x = ConvertToWordArray(sMessage)
```

```
        a = &H67452301
        b = &HEFCDAB89
        c = &H98BADCFE
        d = &H10325476
        For k = 0 To UBound(x) Step 16
    AA = a
    BB = b
    CC = c
    DD = d

    md5_FF a, b, c, d, x(k + 0), S11, &HD76AA478
    md5_FF d, a, b, c, x(k + 1), S12, &HE8C7B756
    md5_FF c, d, a, b, x(k + 2), S13, &H242070DB
    md5_FF b, c, d, a, x(k + 3), S14, &HC1BDCEEE
    md5_FF a, b, c, d, x(k + 4), S11, &HF57C0FAF
    md5_FF d, a, b, c, x(k + 5), S12, &H4787C62A
    md5_FF c, d, a, b, x(k + 6), S13, &HA8304613
    md5_FF b, c, d, a, x(k + 7), S14, &HFD469501
    md5_FF a, b, c, d, x(k + 8), S11, &H698098D8
    md5_FF d, a, b, c, x(k + 9), S12, &H8B44F7AF
    md5_FF c, d, a, b, x(k + 10), S13, &HFFFF5BB1
    md5_FF b, c, d, a, x(k + 11), S14, &H895CD7BE
    md5_FF a, b, c, d, x(k + 12), S11, &H6B901122
    md5_FF d, a, b, c, x(k + 13), S12, &HFD987193
    md5_FF c, d, a, b, x(k + 14), S13, &HA679438E
    md5_FF b, c, d, a, x(k + 15), S14, &H49B40821

    md5_GG a, b, c, d, x(k + 1), S21, &HF61E2562
    md5_GG d, a, b, c, x(k + 6), S22, &HC040B340
    md5_GG c, d, a, b, x(k + 11), S23, &H265E5A51
    md5_GG b, c, d, a, x(k + 0), S24, &HE9B6C7AA
    md5_GG a, b, c, d, x(k + 5), S21, &HD62F105D
    md5_GG d, a, b, c, x(k + 10), S22, &H2441453
    md5_GG c, d, a, b, x(k + 15), S23, &HD8A1E681
    md5_GG b, c, d, a, x(k + 4), S24, &HE7D3FBC8
    md5_GG a, b, c, d, x(k + 9), S21, &H21E1CDE6
    md5_GG d, a, b, c, x(k + 14), S22, &HC33707D6
    md5_GG c, d, a, b, x(k + 3), S23, &HF4D50D87
    md5_GG b, c, d, a, x(k + 8), S24, &H455A14ED
    md5_GG a, b, c, d, x(k + 13), S21, &HA9E3E905
    md5_GG d, a, b, c, x(k + 2), S22, &HFCEFA3F8
    md5_GG c, d, a, b, x(k + 7), S23, &H676F02D9
    md5_GG b, c, d, a, x(k + 12), S24, &H8D2A4C8A
```

```
md5_HH a, b, c, d, x(k + 5), S31, &HFFFA3942
md5_HH d, a, b, c, x(k + 8), S32, &H8771F681
md5_HH c, d, a, b, x(k + 11), S33, &H6D9D6122
md5_HH b, c, d, a, x(k + 14), S34, &HFDE5380C
md5_HH a, b, c, d, x(k + 1), S31, &HA4BEEA44
md5_HH d, a, b, c, x(k + 4), S32, &H4BDECFA9
md5_HH c, d, a, b, x(k + 7), S33, &HF6BB4B60
md5_HH b, c, d, a, x(k + 10), S34, &HBEBFBC70
md5_HH a, b, c, d, x(k + 13), S31, &H289B7EC6
md5_HH d, a, b, c, x(k + 0), S32, &HEAA127FA
md5_HH c, d, a, b, x(k + 3), S33, &HD4EF3085
md5_HH b, c, d, a, x(k + 6), S34, &H4881D05
md5_HH a, b, c, d, x(k + 9), S31, &HD9D4D039
md5_HH d, a, b, c, x(k + 12), S32, &HE6DB99E5
md5_HH c, d, a, b, x(k + 15), S33, &H1FA27CF8
md5_HH b, c, d, a, x(k + 2), S34, &HC4AC5665

md5_II a, b, c, d, x(k + 0), S41, &HF4292244
md5_II d, a, b, c, x(k + 7), S42, &H432AFF97
md5_II c, d, a, b, x(k + 14), S43, &HAB9423A7
md5_II b, c, d, a, x(k + 5), S44, &HFC93A039
md5_II a, b, c, d, x(k + 12), S41, &H655B59C3
md5_II d, a, b, c, x(k + 3), S42, &H8F0CCC92
md5_II c, d, a, b, x(k + 10), S43, &HFFEFF47D
md5_II b, c, d, a, x(k + 1), S44, &H85845DD1
md5_II a, b, c, d, x(k + 8), S41, &H6FA87E4F
md5_II d, a, b, c, x(k + 15), S42, &HFE2CE6E0
md5_II c, d, a, b, x(k + 6), S43, &HA3014314
md5_II b, c, d, a, x(k + 13), S44, &H4E0811A1
md5_II a, b, c, d, x(k + 4), S41, &HF7537E82
md5_II d, a, b, c, x(k + 11), S42, &HBD3AF235
md5_II c, d, a, b, x(k + 2), S43, &H2AD7D2BB
md5_II b, c, d, a, x(k + 9), S44, &HEB86D391

a = AddUnsigned(a, AA)
b = AddUnsigned(b, BB)
c = AddUnsigned(c, CC)
d = AddUnsigned(d, DD)
    Next

    'MD5 = LCase(WordToHex(a) & WordToHex(b) & WordToHex(c) & WordToHex(d))
    MD5=LCase(WordToHex(d) & WordToHex(a))
```

```
End Function
%>
```

(2) 将 MP5.asp 保存至 C:\Inetpub\wwwroot 文件夹中。

4. 创建验证显示文件

完成数据库、数据库连接文件和用户管理系统的密码加密文件的创建后，用户可以参考下面实例的操作，创建本节实例所需的 Function.asp、Safecode.asp、Canvas.asp 和 Font.asp 等 4 个文件。这些文件可以完成用户登录界面中验证码图像的生成与显示。

【练习 13-4】创建用于生成与显示用户登录界面中验证码图像的 Function.asp、Safecode.asp、Canvas.asp 和 Font.asp 文件。

(1) 创建 Function.asp 文件，其代码如下所示：

```
<%
Sub GetSafeCode
 Dim test,Result
 On Error Resume Next
 Set test=Server.CreateObject("Adodb.Stream")
 Set test=Nothing
 If Err Then
        Dim zNum
        Randomize timer
        zNum = cint(8999*Rnd+1000)
        Session("SafeCode") = zNum
        Result = Session("SafeCode")
 Else
 Result = "<img src=""Safecode.asp"" align=""absmiddle"">"
 End If
 Response.Write Result
End Sub
%>
```

(2) 将 Function.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(3) 创建 Safecode.asp 文件，其代码如下所示：

```
<!--#include file="canvas.asp"-->
<%
 Dim objCanvas
 Dim PointX,PointY,PointColor
 Dim iTemp
 Dim SafeCode
 Dim R,G,B,cc,kk
 cc=80
```

```
kk=30
SafeCode = ""
Session("SafeCode") = ""
BGColor = "FFFFFF"
R = Mid(BGColor,1,2)
G = Mid(BGColor,3,2)
B = Mid(BGColor,5,2)
R = DecHex(R)
G = DecHex(G)
B = DecHex(B)
Set objCanvas = New Canvas
objCanvas.GlobalColourTable(0) = RGB(255,255,255) ' White
objCanvas.GlobalColourTable(1) = RGB(0,0,0) ' Black
objCanvas.GlobalColourTable(2) = RGB(255,0,0) ' Red
objCanvas.GlobalColourTable(3) = RGB(0,255,0) ' Green
objCanvas.GlobalColourTable(4) = RGB(0,0,255) ' Blue
objCanvas.GlobalColourTable(5) = RGB(128,0,0)
objCanvas.GlobalColourTable(6) = RGB(0,128,0)
objCanvas.GlobalColourTable(7) = RGB(0,0,128)
objCanvas.GlobalColourTable(8) = RGB(128,128,0)
objCanvas.GlobalColourTable(9) = RGB(0,128,128)
objCanvas.GlobalColourTable(10) = RGB(128,0,128)
objCanvas.GlobalColourTable(11) = RGB(R,G,B)
objCanvas.BackgroundColourIndex = 11
objCanvas.Resize cc,kk,false
Randomize timer
SafeCode = cint(8999*Rnd+1000)
    For iTemp = 0 To 30
    Randomize timer
    PointX = Int(Rnd * cc)
    PointY = Int(Rnd * kk)
    PointColor = Int(Rnd * 3)+2
    objCanvas.ForegroundColourIndex      = PointColor
    objCanvas.Line PointX,PointY,PointX,PointY
    next
objCanvas.ForegroundColourIndex = 1
objCanvas.Line 1,1,cc,1
objCanvas.Line 1,kk,1,1
objCanvas.Line 1,kk,cc,kk
objCanvas.Line cc,1,cc,kk
Session("SafeCode") = SafeCode
dim sc,sk
Randomize timer
```

```
sc = cint(24*Rnd)
sk = cint(11*Rnd)
objCanvas.DrawTextWE sc,sk,SafeCode
objCanvas.Write
Function DecHex (HStr)
Dim Result
Dim i,L
Result = 0
L = Len(Hstr)
For i = L-1 To 0 Step-1
Result = Result + (16 ^ i)*GetDecBit(Mid(HStr,i+1,1))
Next
DecHex = Result
End Function
Function GetDecBit (HStr)
Dim Result
Dim R(16)
Dim i
Result = 0
R(0) = "0"
R(1) = "1"
R(2) = "2"
R(3) = "3"
R(4) = "4"
R(5) = "5"
R(6) = "6"
R(7) = "7"
R(8) = "8"
R(9) = "9"
R(10) = "A"
R(11) = "B"
R(12) = "C"
R(13) = "D"
R(14) = "E"
R(15) = "F"
For i = 0 To 15
if HStr=R(i) Then Result = i : Exit For
Next
GetDecBit = Result
End Function
%>
```

(4) 将 Safecode.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(5) 创建 Canvas.asp 文件，其代码如下所示:

```
<%
' The font pack is included seperately so custom packs can be used
%>
<!--#include file="font.asp"-->
<%
***********************************************************
 ************************* ASPCanvas *************************
 ***********************************************************
     Drawing and presentation object for ASP
     Chris Read (aka Centurix/askdaquack/captainscript)
     Thanks to Richard Deeming (www.trinet.co.uk) for improving
     the arc drawing algorithm
     Thanks to Daniel Hasan for bezier curve adjustments
     Thanks to Tony Stefano for his extra font packs
     Updated 23/02/2003
     ASPCanvas home: http://users.bigpond.net.au/mrjolly/
 ***********************************************************
 This file contains the following classes
 Canvas - Main GIF rendering class
 PixelStack - Used to store an order of pixels
 Point - A single pixel coord
 This file contains the following utility functions
 MakeWord - Convert the value to a big-endian word
 MakeByte - Trim value to an 8 bit value
 Blue - Extract Blue value from RGB
 Green - Extract Green value from RGB
 Red - Extract Red value from RGB
 Low - Retrieve the low 8 bits from the value
 High - Retrieve the high 8 bits from the value
 ShiftLeft - Shift the value left x bits
 ShiftRight - Shift the value right x bits
 This class requires font.asp for text rendering support
 !!!Please read notes.htm for information on using this class!!!
 ***********************************************************
 ASPCanvas Copyright (c) 2002, Chris Read. All rights reserved.
 ***********************************************************
 Redistribution and use in source form, with or without modification,
 are permitted provided that the following conditions are met:
 * Redistributions of source code must retain the above copyright notice,
 this list of conditions and the following disclaimer.
 * All advertising materials mentioning features or use of this software
```

```
must display the following acknowledgement: This product includes software
developed by Chris Read with portions contributed by Richard Deeming,
Daniel Hasan and Tony Stefano.
………(参见本章素材文件)
```

(6) 将 Canvas.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

(7) 创建 Font.asp 文件，其代码如下所示:

```
<%
 **********************************************************
' Definitions of chars 0~9
' **********************************************************
' Font and Letter must be defined to work correctly
Dim Font
Dim C(178)
Dim Letter(19)
Dim TempColor,Fi
'for Fi = 0 to 177
 'Randomize
 'TempColor=Int(Rnd * 6) + 5
 'C(Fi) = TempColor
'next
dim   cd
Set Font = Server.CreateObject("Scripting.Dictionary")
Letter(0) = "00000000000000"
Letter(1) = "00001111100000"
Letter(2) = "00011111110000"
Letter(3) = "00111000111000"
Letter(4) = "00110000011100"
Letter(5) = "01110000001100"
Letter(6) = "01100000001110"
Letter(7) = "01100000001110"
Letter(8) = "11100000001110"
Letter(9) = "11000000001110"
Letter(10) = "11000000001110"
Letter(11) = "11100000001110"
Letter(12) = "11100000001100"
Letter(13) = "11100000001100"
Letter(14) = "01100000001100"
Letter(15) = "01110000011100"
Letter(15) = "00111000011000"
Letter(16) = "00011111110000"
Letter(17) = "00001111100000"
Letter(18) = "0000000000000"
```

```
Font.Add "0",Letter

Letter(0) = "00000000000000"
Letter(1) = "00000001110000"
Letter(2) = "00000001110000"
Letter(3) = "00000011100000"
Letter(4) = "00000011000000"
Letter(5) = "00000011000000"
Letter(6) = "00000011000000"
Letter(7) = "00000111000000"
Letter(8) = "00000111000000"
Letter(9) = "00000111000000"
Letter(10) = "00000110000000"
Letter(11) = "00000110000000"
Letter(12) = "00000110000000"
Letter(13) = "00000110000000"
Letter(14) = "00000110000000"
Letter(15) = "00000110000000"
Letter(15) = "00000110000000"
Letter(16) = "00000110000000"
Letter(17) = "00000010000000"
Letter(18) = "00000000000000"

Font.Add "1",Letter

Letter(0) = "00000000000000"
Letter(1) = "00001111110000"
Letter(2) = "00011111111000"
Letter(3) = "00111000011100"
Letter(4) = "01110000011100"
Letter(5) = "01110000011000"
Letter(6) = "01100000011000"
Letter(7) = "00000000111000"
Letter(8) = "00000001110000"
Letter(9) = "00000001110000"
Letter(10) = "00000011000000"
Letter(11) = "00000111000000"
Letter(12) = "00001110000000"
Letter(13) = "00011000000000"
Letter(14) = "00011000000000"
Letter(15) = "00110000011100"
Letter(16) = "01101111111100"
```

```
Letter(17) = "01111111111110"
Letter(18) = "01111100000000"
Letter(19) = "00000000000000"

Font.Add "2",Letter

Letter(0) = "00000000000000"
Letter(1) = "00001111111000"
Letter(2) = "00111111111000"
Letter(3) = "01110000111100"
Letter(4) = "01100000011000"
Letter(5) = "01000000111000"
Letter(6) = "00000000111000"
Letter(7) = "00000001110000"
Letter(8) = "00000011000000"
Letter(9) = "00000111110000"
Letter(10) = "00000100111000"
Letter(11) = "00000000011100"
Letter(12) = "00000000011100"
Letter(13) = "00000000011100"
Letter(14) = "00000000011100"
Letter(15) = "00000000011000"
Letter(16) = "11100000111000"
Letter(17) = "11111111110000"
Letter(18) = "01111111100000"
Letter(19) = "00000000000000"

Font.Add "3",Letter

Letter(0) = "00000000000000"
Letter(1) = "00000000111000"
Letter(2) = "00000001111000"
Letter(3) = "00000011100000"
Letter(4) = "00000111011100"
Letter(5) = "00001110011100"
Letter(6) = "00001100011000"
Letter(7) = "00011000111000"
Letter(8) = "00111000110000"
Letter(9) = "01110000110000"
Letter(10) = "01100000110000"
Letter(11) = "01100000110000"
Letter(12) = "11000111111110"
Letter(13) = "11111111111100"
```

```
Letter(14) = "11111111100000"
Letter(15) = "11100001100000"
Letter(16) = "00000001110000"
Letter(17) = "00000000110000"
Letter(18) = "00000000110000"
Letter(19) = "00000000100000"

Font.Add "4",Letter

Letter(0) = "00000000000000"
Letter(1) = "00001100000100"
Letter(2) = "00011111111110"
Letter(3) = "00011111111100"
Letter(4) = "00011110000000"
Letter(5) = "00011000000000"
Letter(6) = "00111000000000"
Letter(7) = "00111000000000"
Letter(8) = "00111111110000"
Letter(9) = "00111111111000"
Letter(10) = "00000000011000"
Letter(11) = "00000000011000"
Letter(12) = "00000000011000"
Letter(13) = "00000000011000"
Letter(14) = "00000000011000"
Letter(15) = "00000000011000"
Letter(16) = "00000001111000"
Letter(17) = "01111111110000"
Letter(18) = "00111111000000"
Letter(19) = "00000000100000"

Font.Add "5",Letter

Letter(0) = "00000000000000"
Letter(1) = "00000011110000"
Letter(2) = "00000111100000"
Letter(3) = "00001110000000"
Letter(4) = "00011100000000"
Letter(5) = "00111000000000"
Letter(6) = "00110000000000"
Letter(7) = "00110000000000"
Letter(8) = "01111111110000"
Letter(9) = "01111111111000"
Letter(10) = "01110000011100"
```

```
Letter(11) = "01100000001100"
Letter(12) = "01100000001100"
Letter(13) = "01100000001100"
Letter(14) = "01100000001100"
Letter(15) = "01110000011100"
Letter(16) = "00110000011100"
Letter(17) = "00111111111000"
Letter(18) = "00011111110000"
Letter(19) = "00000000000000"

Font.Add "6",Letter

Letter(0) = "00000000000000"
Letter(1) = "00100111111110"
Letter(2) = "01111111111100"
Letter(3) = "01111110011100"
Letter(4) = "00000000011000"
Letter(5) = "00000000111000"
Letter(6) = "00000000110000"
Letter(7) = "00000000110000"
Letter(8) = "00000000110000"
Letter(9) = "00000001110000"
Letter(10) = "00000001100000"
Letter(11) = "00000001100000"
Letter(12) = "00000001100000"
Letter(13) = "00000001100000"
Letter(14) = "00000011100000"
Letter(15) = "00000011100000"
Letter(16) = "00000011100000"
Letter(17) = "00000001000000"
Letter(18) = "00000001000000"
Letter(19) = "00000000000000"

Font.Add "7",Letter

Letter(0) = "00000000000000"
Letter(1) = "00001111110000"
Letter(2) = "00011111111000"
Letter(3) = "00111000011000"
Letter(4) = "00110000011000"
Letter(5) = "01110000011100"
Letter(6) = "01110000011000"
Letter(7) = "00110000011000"
```

```
Letter(8) = "00111101111000"
Letter(9) = "00011111111000"
Letter(10) = "00111000111100"
Letter(11) = "01110000001100"
Letter(12) = "01110000001100"
Letter(13) = "01100000001110"
Letter(14) = "01100000001100"
Letter(15) = "01100000001100"
Letter(16) = "01110000011100"
Letter(17) = "00111111111100"
Letter(18) = "00011111110000"
Letter(19) = "00000000000000"

Font.Add "8",Letter

Letter(0) = "00000000000000"
Letter(1) = "00011111110000"
Letter(2) = "00111111111000"
Letter(3) = "01110000111000"
Letter(4) = "01110000011100"
Letter(5) = "01100000001100"
Letter(6) = "01100000001100"
Letter(7) = "01100000001100"
Letter(8) = "01100000001100"
Letter(9) = "01110000011100"
Letter(10) = "00111111111100"
Letter(11) = "00011111111100"
Letter(12) = "00000000011000"
Letter(13) = "00000000011000"
Letter(14) = "0000000011100"
Letter(15) = "00000001110000"
Letter(16) = "00000011100000"
Letter(17) = "00000111000000"
Letter(18) = "00011110000000"
Letter(19) = "00000000000000"
Font.Add "9",Letter
%>
```

(8) 将 Font.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

5. 创建用户登录文件

进入用户管理系统时，首先进入到如图 13-3 所示的 Chkwd.asp 页面。在 Chkwd.asp 页面文件中，为了保证用户登录后，不再重复显示登录窗口，需要设置一个 Session 变量

Passed。如果 Session(Passed)=True，表示用户已经登录成功；反之，如果 Session(Passed)= False，表示用户还没有登录，则显示用户登录窗口。

【练习 13-5】创建用户管理系统的用户登录文件 Chkpwd.asp。

(1) 创建用户登录页面文件 Chkpwd.asp，其代码如下所示：

```
<!--#include file = "conn.asp"-->
<!--#include file = "md5.asp"-->
<!--#include file = "function.asp"-->
<%
   '如果尚未定义 Passed 对象，则将其定义为 False，表示没有通过身份认证
   if IsEmpty(Session("Passed")) then
      Session("Passed") = False
   End if
  '如果 Session("Passed") = False，表示没有通过身份认证
  if Session("Passed") = False then
  '读取从表单传递过来的身份数据
    User_Name = Trim(Request.form("User_Name"))
    User_Pwd = Request.form("User_Pwd")

     if User_Name = "" or User_Pwd = "" then
        Errmsg = "请输入用户名和密码登录用户管理系统"
     else
         If not IsNumeric(Request.Form("confirm")) Then
          Response.Write "<script>alert('你输入的验证码为非数字！ ');</script>"
          Response.End
        End If
        If (int(Session("SafeCode"))=int(Request.Form("confirm"))) Then
           set rs = server.createobject("adodb.recordset")
           sql = "Select * from users where User_Name = '"&User_Name&"'"
           rs.open sql,conn,1,3
           if rs.EOF then
                Errmsg = "用户不存在"
                conn.Close
                Set conn = nothing
                 else
                if md5(User_Pwd)<>rs.Fields("User_Pwd") then
                      Errmsg = "密码不正确"
                      conn.Close
                      Set conn = nothing
                else
                      Errmsg = ""
                      Session("Passed") = True
                      Session("User_Name") = rs.Fields("User_Name")
```

```
                    Session("Id") = rs.Fields("Id")
                end if
            end if
        Else
            Errmsg = "验证码输入错误！"
      End If
    end if
    end if
    if Not Session("Passed") then
  %>
<html>
<head>
<body bgcolor="teal"text="red">
<title>请输入用户名和密码登录用户管理系统</title>
</head>
<body>
<script language = "VBScript">
Function ChkFields()
    If document.MyForm.User_Name.value ="" Then
            window.alert("请输入用户名！")
            return false
    End If
    If document.MyForm.User_Pwd.value ="" Then
            window.alert("请输入密码！")
            return false
    End If
    If document.MyForm.confirm.value = "" Then
            window.alert("请输入验证码！")
        return false
    End If
    return true
End Function
</script>
<p align = "center"><font color = "#0000ff" size = 5 >用户管理系统(身份验证页面)</font></p>
<p align = "center"><font color = "#800000">
<%=Errmsg%>
</font></p>
<form method = "post" action ="<%=request.servervariables("PATH_INFO")%>" name = "MyForm"
        onsubmit = "return ChkFields()">
<center>
<table border="0" width="40%">
  <tr>
    <td width="27%" bgcolor="#C0C0C0" align="center"><font size="2">用户名：</font></td>
    <td width="73%"><input type = "text" name = "User_Name" size = 20></td>
```

```
        </tr>
        <tr>
          <td width="27%" bgcolor="#C0C0C0" align="center"><font size="2">输入密码：</font></td>
          <td width="73%"><input type = "password" name = "User_Pwd" size = 20></td>
        </tr>
        <tr>
          <td width="27%" bgcolor="#C0C0C0" align="center"><font size="2">验证码：</font></td>
          <td width="73%"><input type = "text" name = "confirm" size = 10><%call
                GetSafeCode()%></td>
        </tr>
      </table>
      <p align = "center"><input type = "submit" value = "提交" name = "B1">  
      <input type = "reset" value = "重置" name = "B2"></p>
      </form>
      <p align = "center"></p>
      </body>
      </html>
      <%
      response.end
      End if
      %>
```

(2) 选择“文件”|“另存为”命令，将上面输入的代码以文件名 Chkpwd.asp 保存至 C:\Inetpub\wwwroot 文件夹中。

6. 创建管理系统界面

完成以上的一系列操作后，用户可以参考下面实例的操作创建用户管理系统的主界面。该界面能够根据用户不同的权限显示不同的系统界面。例如，当用户以系统管理员 Admin 登录时，Index.asp 页面显示如图 13-11 所示的页面，在该页面中系统管理员可以使用“用户管理”、“修改密码”和“退出登录”等命令；而当用户以普通用户身份登录时，Index.asp 页面则会显示如图 13-12 所示的页面，在该页面中用户只可以使用“修改密码”和“退出登录”命令。

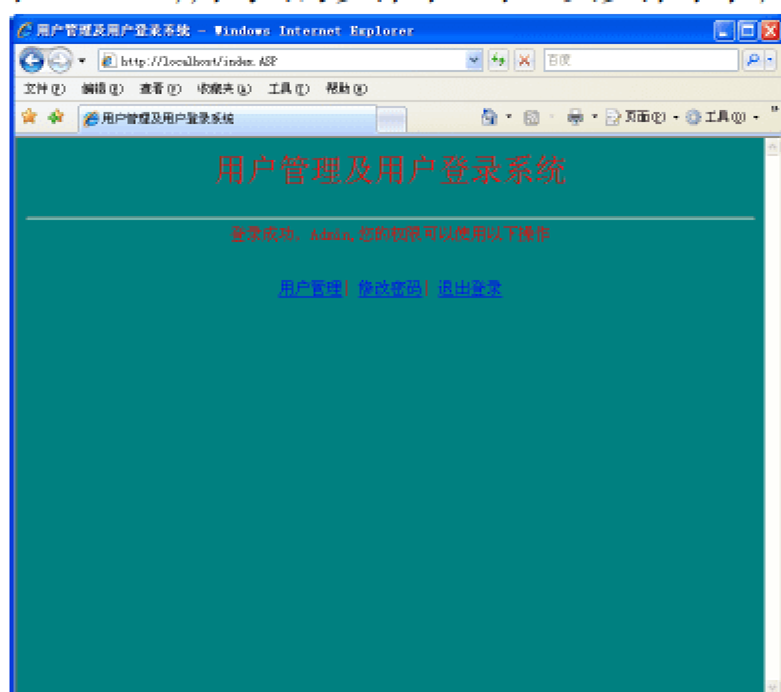


图 13-11　Admin 登录后的 Index.asp 页面

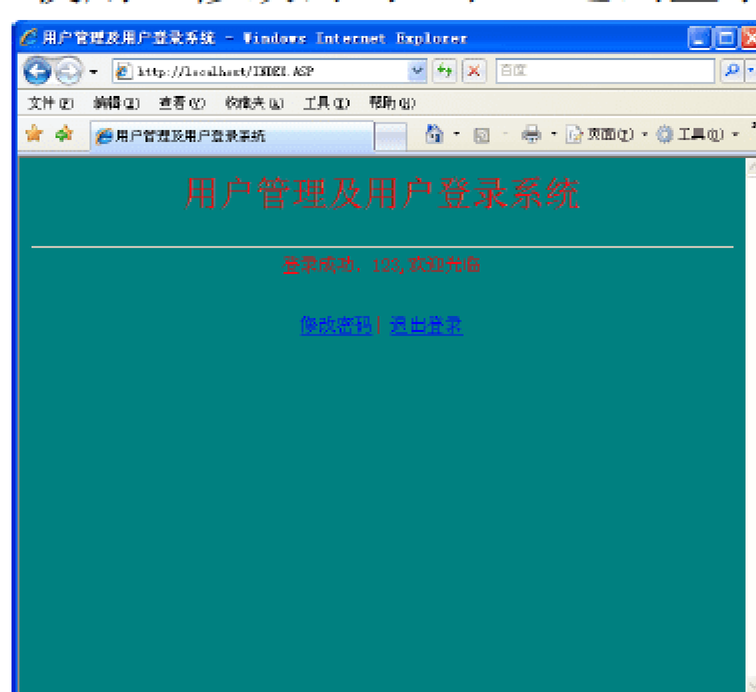


图 13-12　普通用户登录成功后的 Index.asp 页面

【练习 13-6】创建用户管理系统的管理主界面 Index.asp。

(1) 创建用户管理系统的管理主界面 Index.asp，其代码如下所示：

```
<!--#include file= "ChkPwd.asp"-->
<html>

<head>
<title>用户管理及用户登录系统</title>
</head>
<body bgcolor="teal"text="red">
<body>

<p align="center"><font size="6">用户管理及用户登录系统</font></p>
<hr>
<center>
<script language = "VBScript">
Function newwin(url)
  newwin = window.open(url,"newwin","toolbar = no,location = no,directiories = no,status =
      no,menubar = no,scrollbars = yes,resizable = yes,width = 400,height = 380")
  newwin.focus()
  return false
End Function
</script>
<%
  '如果登录成功，则根据用户类型决定显示内容
  if session("Passed") = True then
    response.write("登录成功，"&Session("User_Name")&",您的权限可以使用以下操作
      <br><br><br><center>")
    if session("User_Name") = "Admin" then
      Response.write("<a href = UserList.asp>用户管理</a>|    <a href =
        pwdchange.asp?UserId="&Session("Id")&" onclick = ""return newwin(this.href)"">修改密
        码</a>|    <a href = logout.asp>退出登录</a>")
    else
        response.write("<a href = pwdchange.asp?UserId="&Session("Id")&" onclick = ""return
          newwin(this.href)"">修改密码</a>|    <a href = logout.asp>退出登录</a>")
  end if
  end if
%>
</body>
</html>
```

(2) 将 Index.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

7. 创建用户管理页面

用户管理页面是系统管理员 Admin 登录用户管理系统后，在如图 13-11 所示的页面中单击“用户管理”链接打开的页面。在该页面的用户列表中，列出了用户管理系统数据库中所有的用户记录，每条记录包括用户姓名、电子邮箱和修改、删除等操作。在用户列表的下方，还有一个“添加用户信息”链接，系统管理员可以通过该链接进入添加用户页面，如图 13-13 所示。



单击“用户管理”链接　　　　用户列表页面

图 13-13　用户管理页面

【练习 13-7】创建系统管理员专用的用户管理页面 UserList.asp。

(1) 创建用户管理页面文件 UserList.asp，其代码如下所示。

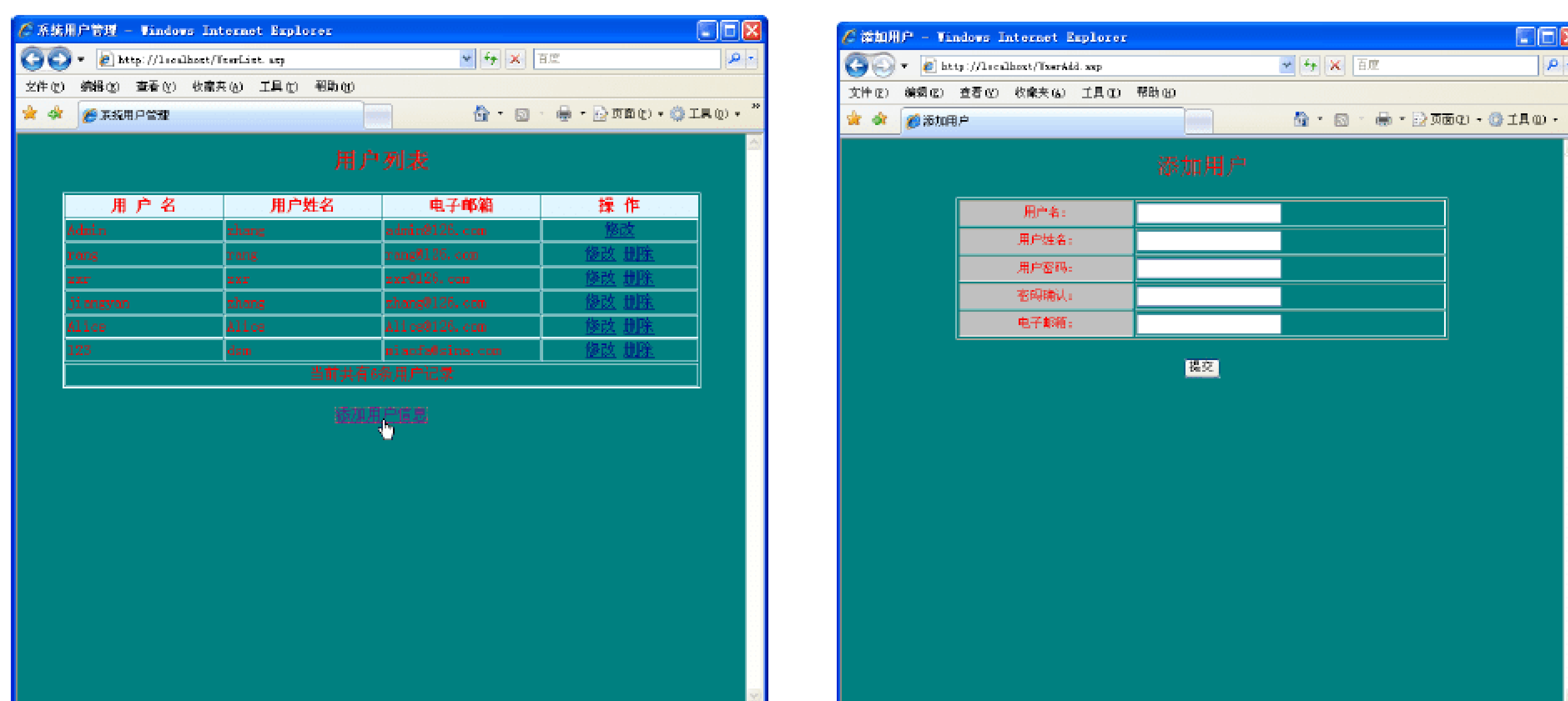
```
<!--#include file= "ChkPwd.asp"-->
<%
If Session("User_Name") <> "Admin" Then    '登录的不是 Admin
 '给出提示，并返回前一页
 Response.Write "<script>alert('您不是系统管理员，没有此权限！');history.back()</script>"
 Session("Passed") = False
Else
%>
<html>
<body bgcolor="teal"text="red">
  <head>
    <title>系统用户管理</title>
    <script language = "vbscript">
    Function newwin(url) '定义 newwin  过程
        '使用 window  的 open  方法打开一个窗口，指定窗口大小，并且大小可变，
        '没有状态栏、地址栏、菜单和工具栏，窗口中显示的内容由 URL 指定
        newwin =window.open(url,"newwin","toolbar=no,location=no,directories=no,status=no, menubar=no,
         scrollbars=yes,resizable=yes,width=400,height=380")
```

```
        newwin.focus()
   return false
   End Function
   </script>
  </head>
  <body link = #000080 vlink = #008080>
   <h3></h3>
   <h2 align=center>用户列表</h2>
   <table width = 90% align = center cellspacing = 1 cellpadding = 2 border=1 bordercolor=#808080
          bordercolordark=#FFFFFF bordercolorlight=#E1F5FF>
     <tr>
       <td align=center width=20% bgcolor=#E1F5FF><b>用  户  名</b></td>
  <td align=center width=20% bgcolor=#E1F5FF><b>用户姓名</b></td>
  <td align=center width=20% bgcolor=#E1F5FF><b>电子邮箱</b></td>
  <td align=center width=20% bgcolor=#E1F5FF><b>操      作</b></td>
     </tr>
  <%
  '定义 RecordSet 对象
  Set rs = Server.CreateObject("ADODB.RecordSet")
  rs.Open "Select * from Users Order by Id",conn,1,3
  rCount = rs.RecordCount
  '循环显示所有的用户数据，同时画出表格
  Do While Not rs.EOF
  %>
   <tr>
  <td><%=rs("User_Name")%></td>
  <td><%=rs("Name")%></td>
  <td><%=rs("Email")%></td>
  <td align=center><a href=UserEdit.asp?userid=<%=rs("Id")%> onclick="return newwin(this.href)">
       修改</a>
  <%If rs("User_Name")<>"Admin" Then%>
        <a href=UserDel.asp?userid=<%=rs("Id")%> onclick="return newwin(this.href)">删除
          </a></td>
  <%End If%>
```

(2) 将 UserList.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

8. 创建添加用户页面

添加用户页面是管理员在 UserList.asp 页面中单击“添加用户信息”链接所打开的页面，在该页面中管理员可以添加用户记录，如图 13-14 所示。



单击“添加用户信息”链接　　　　添加用户页面

图 13-14　打开添加用户页面

【练习 13-8】创建添加用户页面 UserAdd.asp。

(1) 创建用户管理系统的新用户添加页面文件 UserAdd.asp，其代码如下所示：

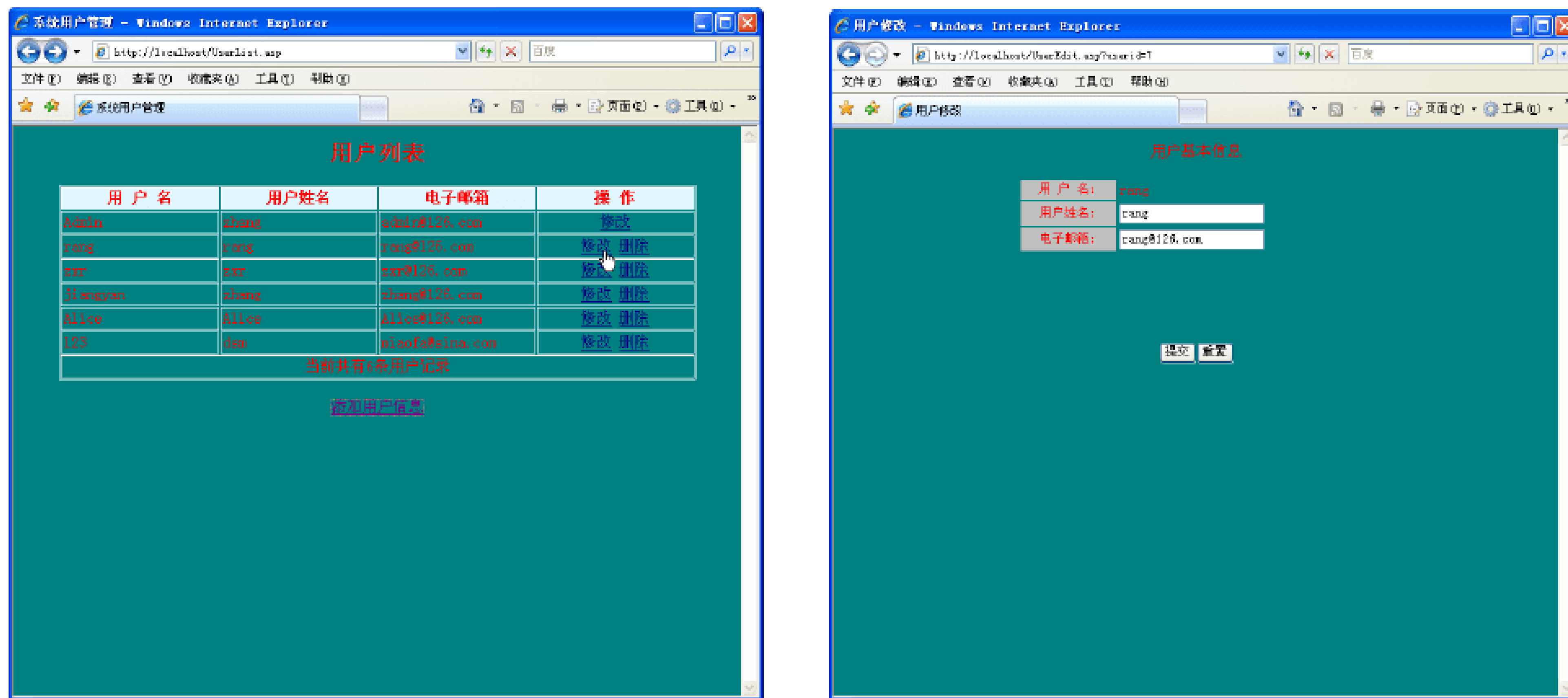
```
</form>
<script language = "VBScript">
function ChkFields()
    if document.myform.User_Name.value = "" Then            '没有输入用户名
        window.alert("请输入用户名")                        '给出提示
        myform.User_Name.focus()                            '将“用户名”文本框设为输入焦点
        return false
    End If
    if document.myform.User_Name.value.Length <= 6 Then     '用户名长度太小
        window.alert("请用户名长度必须大于 2！ ")           '给出提示
        myform.User_Name.focus()                            '将“用户名”文本框设为输入焦点
        return false
    End If
    if document.myform.Pwd.value.Length <= 6 Then           '密码长度小于 6
        window.alert("新密码长度大于 6")                    '给出提示
        myform.User_Name.focus()                            '将“用户名”文本框设为输入焦点
        return false
    End If
    if document.myform.Pwd.value ="" Then                   '没有输入密码
        window.alert("请输入新密码")                        '给出提示
        myform.User_Name.focus()                            '将“用户名”文本框设为输入焦点
        return false
    End If
      if document.myform.Pwd1.value ="" Then                '没有输入确认密码
```

```
            window.alert("请确认新密码")                    '给出提示
            myform.User_Name.focus()                      '将“用户名”文本框设为输入焦点
            return false
        End If
        '两次输入的密码不相同
        if document.myform.Pwd.value <> document.myform.Pwd1.value Then
            window.alert("两次输入的新密码必须相同")          '给出提示
            myform.User_Name.focus()                       '将“用户名”文本框设为输入焦
点
            return false
        End If
        return true
    End Function
    </script>
    </body>
    </html>
    <%
    End If
    %>
```

(2) 将 UserAdd.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

9. 创建修改用户页面

修改用户页面是管理员在 UserList.asp 页面中单击“修改”链接所打开的页面，在该页面中，用户可以对用户管理系统中所有用户的用户资料进行修改，如图 13-15 所示。



单击“修改”链接　　　　修改用户页面

图 13-15　打开修改用户页面

【练习 13-9】创建修改用户页面 UserEdit.asp。

(1) 创建用户管理系统的修改用户页面 UserEdit.asp，其代码如下所示。

```
<!--#include file = "conn.asp"-->
<%
If Session("UserName") <> "Admin" Then
 Response.Write "<script>alert('您不是系统管理员，没有此权限！ ');history.back()</script>"
 Session("Passed") = False
Else
%>
<html>
<body bgcolor="teal"text="red">
<head>
<meta http-equiv="Content-Language" content="zh-cn">
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>用户修改</title>
</head>
<body>
<%
Dim Uid
Uid = Request.QueryString("UserId")
sql = "Select * from Users where Id = "&Uid
'定义 RecordSet 对象
Set rs = Server.CreateObject("ADODB.RecordSet")
rs.Open sql,conn,1,3
If rs.EOF Then
 Response.Write ("<h2>不存在此用户名！ </h2>")
Else
%>
<form method="POST" action="UserSave.asp" name=myform onsubmit="return ChkField()">
 <p align="center">用户基本信息</p>
 <center>
 <table border="0" width="371" id="table1">
        <tr>
               <td bgcolor="#C0C0C0" width="97" align="center"><font size="2">用
                      户  名:</font></td>
               <td width="260"><%=rs("UserName")%><input type = hidden name = UserID value =
                      <%=rs("id")%>></td>
        </tr>
        <tr>
               <td bgcolor="#C0C0C0" width="97" align="center"><font size="2">用户姓
                      名:</font></td>
               <td width="260"><input type=text name=Ename size=20
                      value="<%=rs("Ename")%>"></td>
        </tr>
        <tr>
```

```
            <td bgcolor="#C0C0C0" width="97" align="center"><font size="2">电子邮
                箱:</font></td>
            <td width="260"><input type=text name=Email size=20
                value="<%=rs("Email")%>"></td>
        </tr>
 </table>
 <p>　</p>
 <p>　</p>
 <p align="center"><input type="submit" value="提交" name="B1"><input type="reset" value="重
        置" name="B2"></p>
</form>
<%End If%>
</body>
</html>
<%
End If
%>
```

(2) 将 UserEdit.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

10. 创建删除用户文件

用户管理系统的管理员在 UserList.asp 页面中单击“删除”链接可以打开 UserDel.asp 文件，从数据库中删除某条用户记录。

【练习 13-10】创建用于删除用户记录的 ASP 文件 UserDel.asp。

(1) 创建用户管理系统的删除用户页面 UserDel.asp，其代码如下所示。

```
<!--#include file= "conn.asp"-->
<%
If Session("User_Name") <> "Admin" Then
 Response.Write "<script>alert('您不是系统管理员，没有此权限！');history.back()</script>"
 Session("Passed") = False
Else
%>
<html>
<head>
<title>删除用户信息</title>
</head>
<body>
<%
Dim Uid
'读取 UserId 参数
Uid=Request.QueryString("UserId")
Set Drs = Server.CreateObject("ADODB.RecordSet")
```

```
Drs.Open "Delete from Users where Id = "&Uid,conn,1,3
Response.Write ("<center><h2>成功删除</h2>")
Response.Write "<a href = UserList.asp>返回</a>"
%>
</body>
</html>
<%
End If
%>
```

(2) 选择“文件”|“另存为”命令，将上面输入的代码以文件名 UserDel.asp 保存至 C:\Inetpub\wwwroot 文件夹中。

11. 创建修改用户密码页面

修改用户密码页面是管理员和普通用户修改用户登录密码的页面，在该页面中用户可以修改登录用户管理系统的用户密码，如图 13-16 所示。



单击“修改密码”链接　　　　修改用户密码

图 13-16　打开修改用户密码页面

【练习 13-11】创建修改用户密码页面 PwdChange.asp。

(1) 创建用户管理系统的用户密码修改页面 PwdChange.asp，其代码如下所示。

```
<!--#include file = "ChkPwd.asp"-->
<!--#include file = "conn.asp"-->
<html>
<body bgcolor="teal"text="red">
<head>
<meta http-equiv="Content-Language" content="zh-cn">
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>密码修改</title>
<script language="vbscript">
Function ChkFields()
```

```
    If Document.myform.Pwd.value.length<6 Then
        alert("新密码长度必须大于等于 6！ ")
        myform.Pwd.focus()
        return false
    End If
    If Document.myform.Pwd1.value="" Then
        alert("请确认新密码！ ")
        myform.Pwd.focus()
        return false
    End If
    If Document.myform.Pwd.value<>Document.myform.Pwd1.value Then
        alert("两次输入的新密码必须相等！ ")
        return false
    End If
    return true
End Function
</script>
</head>
<body>
<%
User_ID = Request.QueryString("User_ID")
Set rs = server.CreateObject("ADODB.RecordSet")
rs.Open "Select * from users where id="&User_ID,conn,1,3
%>
<form method="POST" name="myform" action="SavePwd.asp?User_ID=<%=User_ID%>"
      onsubmit="return ChkField()">
 <p>  </p>
 <p align="center">修改密码</p>
 <center>
 <table border="0" width="355" id="table1">
 <tr>
 <td width="100" bgcolor="#C0C0C0" align="center"><font size="2">用
              户  名:</font></td>
     <td width="241"><%=rs("UserName")%>  </td>
     </tr>
     <tr>
     <td width="100" bgcolor="#C0C0C0" align="center"><font size="2">原始密码:</font></td>
     <td width="241"><input type="password" name="OriPwd" size="20"></td>
     </tr>
     <tr>
         <td width="100" bgcolor="#C0C0C0" align="center"><font size="2">新
            密  码:</font></td>
         <td width="241"><input type="password" name="Pwd" size="20"></td>
```

```
</tr>
 <tr>
              <td width="100" bgcolor="#C0C0C0" align="center"><font size="2">密码确
                  认:</font></td>
              <td width="241"><input type="password" name="Pwd1" size="20"></td>
        </tr>
 </table>
 <p>   </p>
 <p>   </p>
 <p align="center"><input type="submit" value="提交" name="B1"><input type="reset" value="重
       置" name="B2"></p>
</form>
<%
rs.Close
Set rs = nothing
%>
</body>
</html>
```

(2) 将 PwdChange.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

12. 创建用户退出登录页面

在 Index.asp 页面中，单击“退出登录”链接，可以打开退出登录页面 Logout.asp，退出用户管理系统。

【练习 13-12】创建退出登录页面 Logout.asp。

(1) 创建用户管理系统的退出登录页面 Logout.asp，其代码如下所示：

```
<%
Session("Passed") = false
Session("User_Name")=""
Response.Redirect "index.asp"
%>
```

(2) 将 Logout.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

13.1.4　实例测试

完成本节实例的操作后启动 IE 浏览器，在地址栏输入 http://localhost/index.asp 并按下 Enter 键显示用户登录界面。在用户登录界面中的“用户名”文本框中输入管理员用户名 admin，在“输入密码”文本框中输入管理员密码 admin，在“验证码”文本框中根据页面中的验证码提示图片中显示的数字输入验证码(如图 13-17 所示)。然后单击“提交”按钮，打开如图 13-18 所示的用户管理界面。



图 13-17　用户登录界面

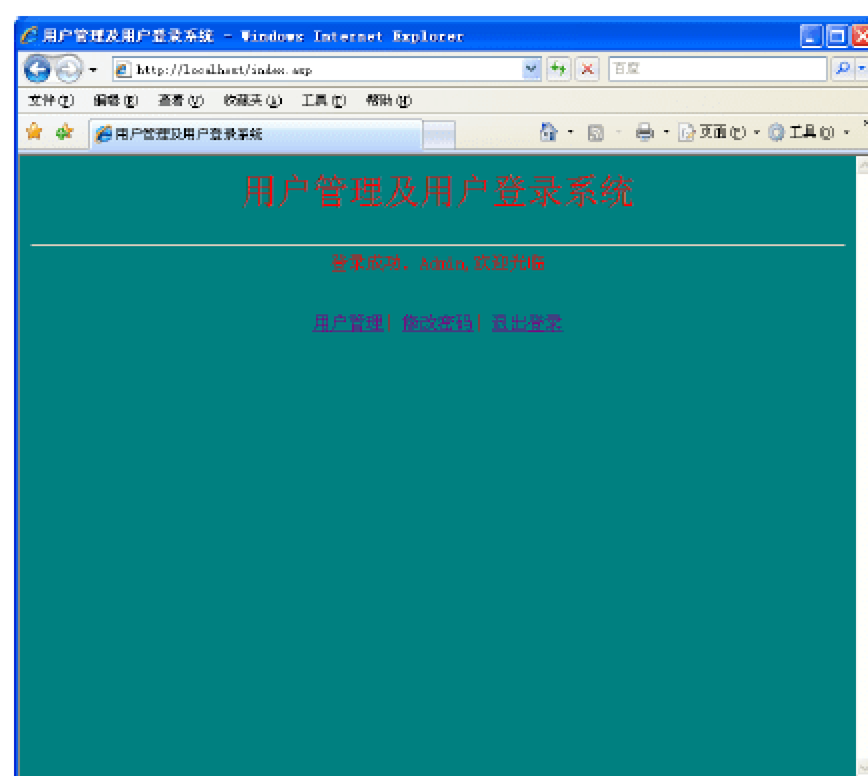


图 13-18　用户管理界面(管理员身份登录)

单击如图 13-12 所示的用户管理界面中的“用户管理”链接，进入如图 13-19 所示的用户管理列表页面，在该页面中用户可以通过单击页面中具体用户名称后的“修改”链接打开如图 13-20 所示的用户资料修改页面，可以修改用户基本信息。

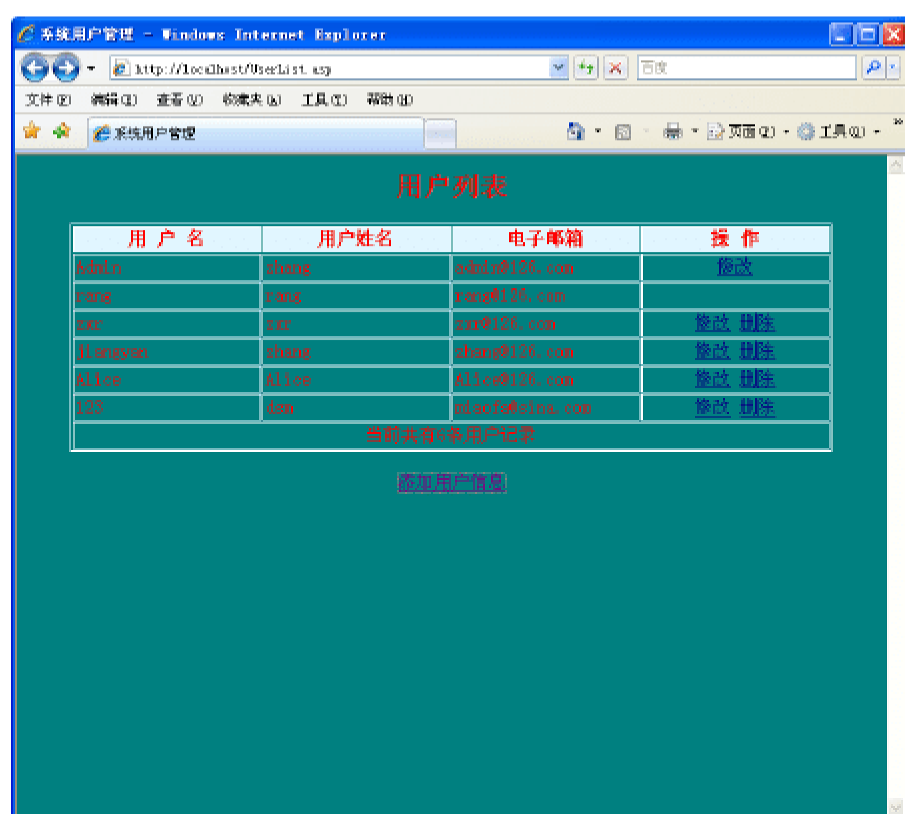


图 13-19　用户管理列表页面

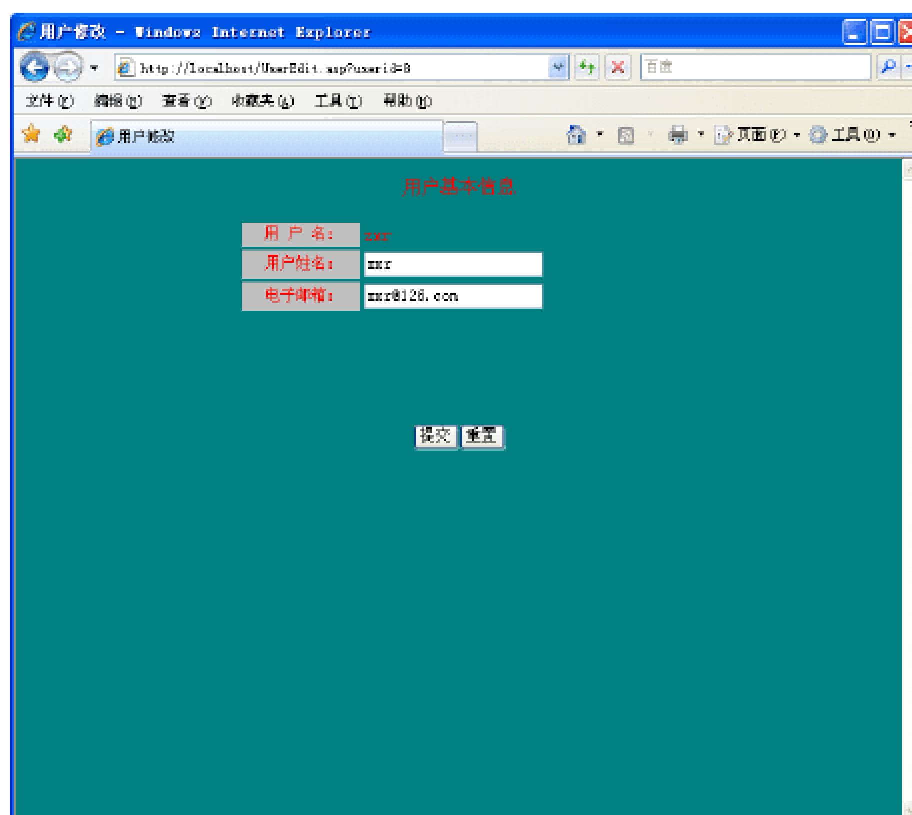


图 13-20　用户资料修改页面

单击如图 13-19 所示的用户管理列表页面中具体用户名称后的“删除”链接，可以将选中的普通用户从用户管理系统中删除，如图 13-21 所示。单击用户管理列表中的“添加用户信息”链接，可以打开如图 13-22 所示的添加用户页面，在用户管理系统中添加新用户。

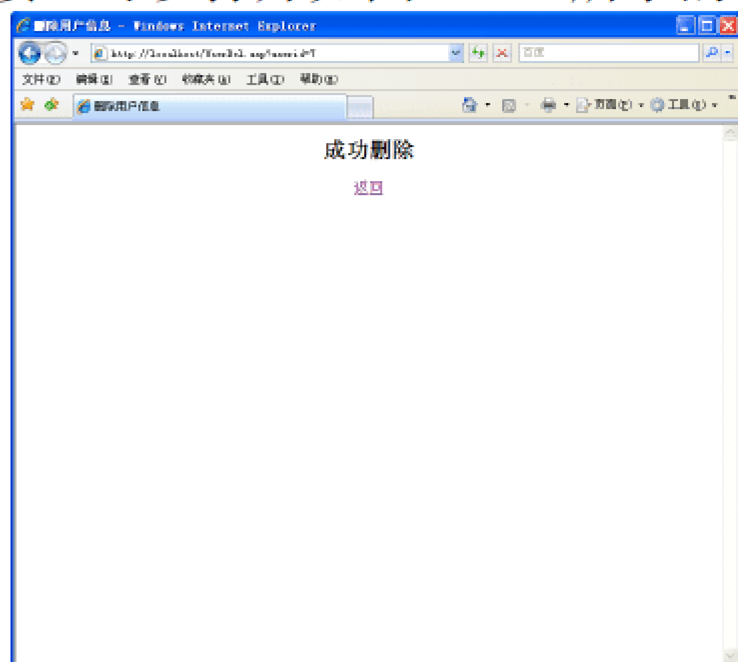


图 13-21　成功删除用户

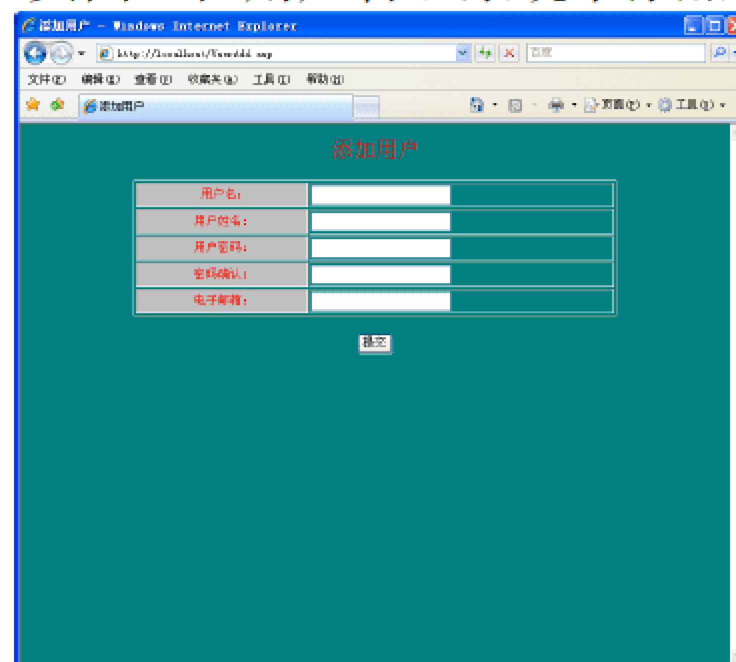


图 13-22　添加用户页面

单击如图 13-18 所示的用户管理界面中的“修改密码”链接，打开如图 13-23 所示的用户密码修改页面可以修改当前登录用户的用户登录密码。单击如图 13-18 所示的用户管理界面中的“退出登录”链接，可以使当前登录用户退出登录，并返回用户管理系统主界面 Index.asp，如图 13-24。

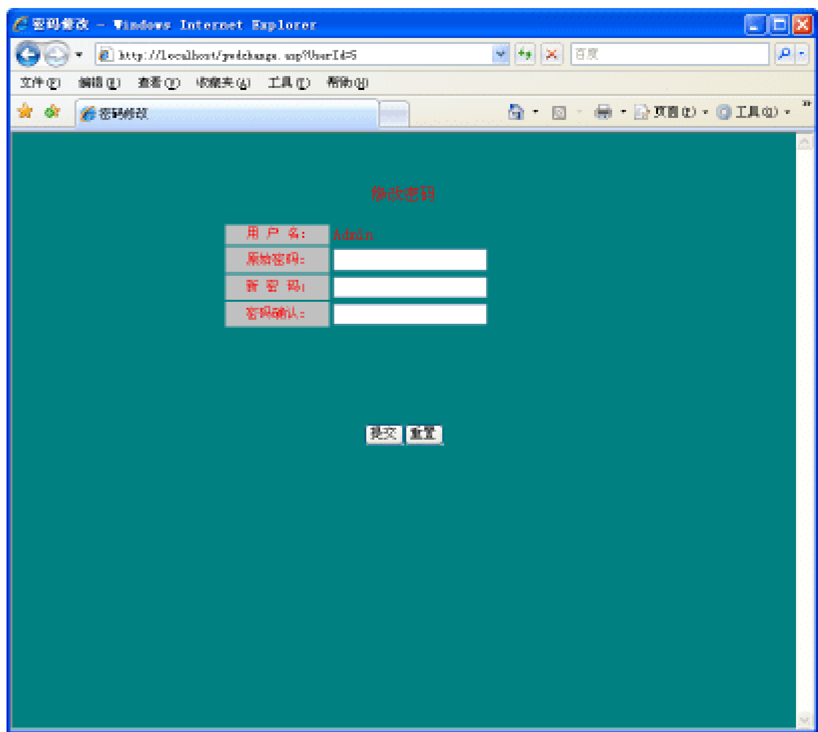


图 13-23　修改用户密码

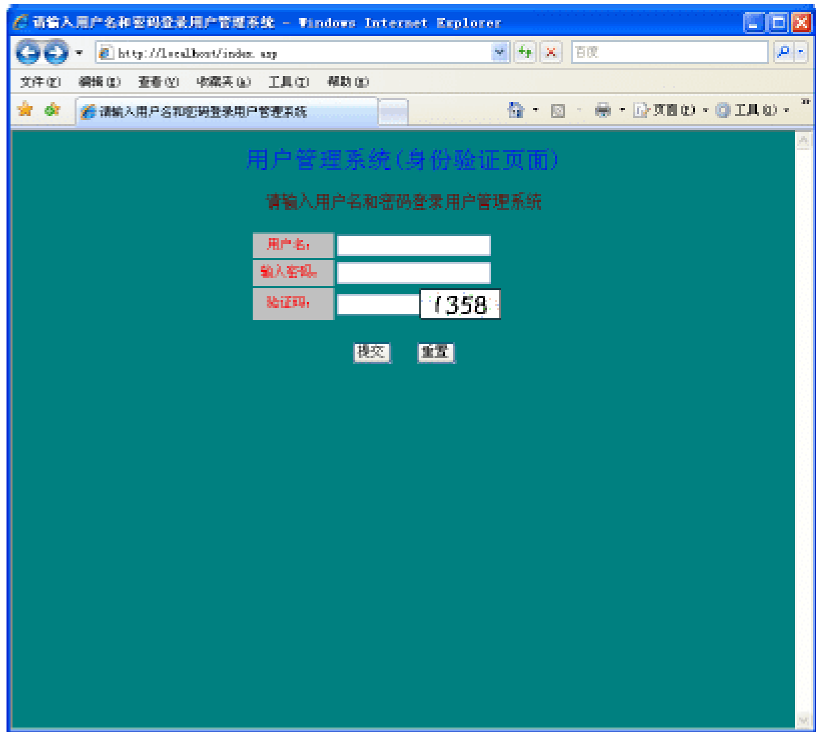


图 13-24　返回 Index.asp 页面

如果用户在用户登录界面中的“用户名”文本框中输入用户名 123，在“输入密码”文本框中输入密码 123，在“验证码”文本框中根据页面中的验证码提示图片中显示的数字输入验证码(如图 13-25 所示)。然后单击“提交”按钮，打开如图 13-26 所示的普通用户管理界面。



图 13-25　输入普通用户登录信息

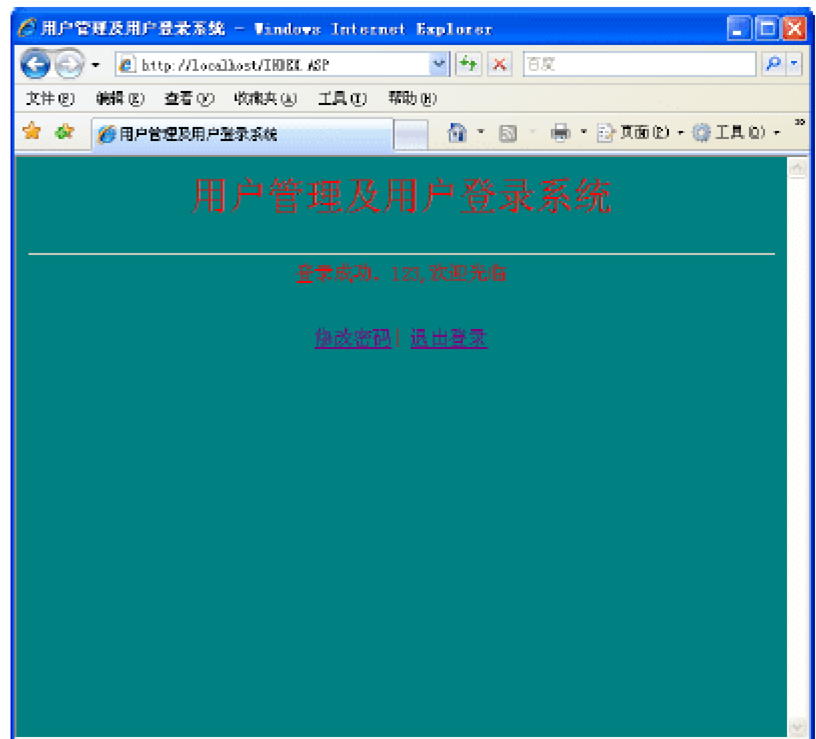


图 13-26　普通用户管理页面

在如图 13-26 所示的普通用户管理页面中，可以单击“修改密码”链接，修改普通用户的密码，单击“退出登录”链接，返回 Index.asp 页面。

13.2　留　言　板

留言板是网站中常见的页面。许多特定的主题站点都提供留言功能，网站的注册用户在登录网站后可以在其中发表文章或留言，其他用户则可以针对这些留言进行回复或发表新的话题。

13.2.1　功能介绍

本节所介绍的留言板模块的主界面如图 13-27 所示。在该界面中可以看到留言板中其他用户的留言和留言板管理员的回复内容。单击留言板主界面中的“发表留言”按钮，还可以打开如图 13-28 所示的留言发表页面，发表新的留言。



图 13-27　留言板主界面



图 13-28　发表留言页面

留言板管理员可以单击留言板主界面右下角的“管理留言板”按钮，打开留言板登录页面登录留言板管理页面。在留言板管理页面中，留言板管理员可以根据需要管理留言板内容。

13.2.2　模块设计

在网站留言板模块中，用户不需要注册就能够发表留言、查看留言。系统可以有多个管理员，每个管理员可以设置不同的回复内容、颜色和回复人提示，其主要结构如图 13-29 所示。

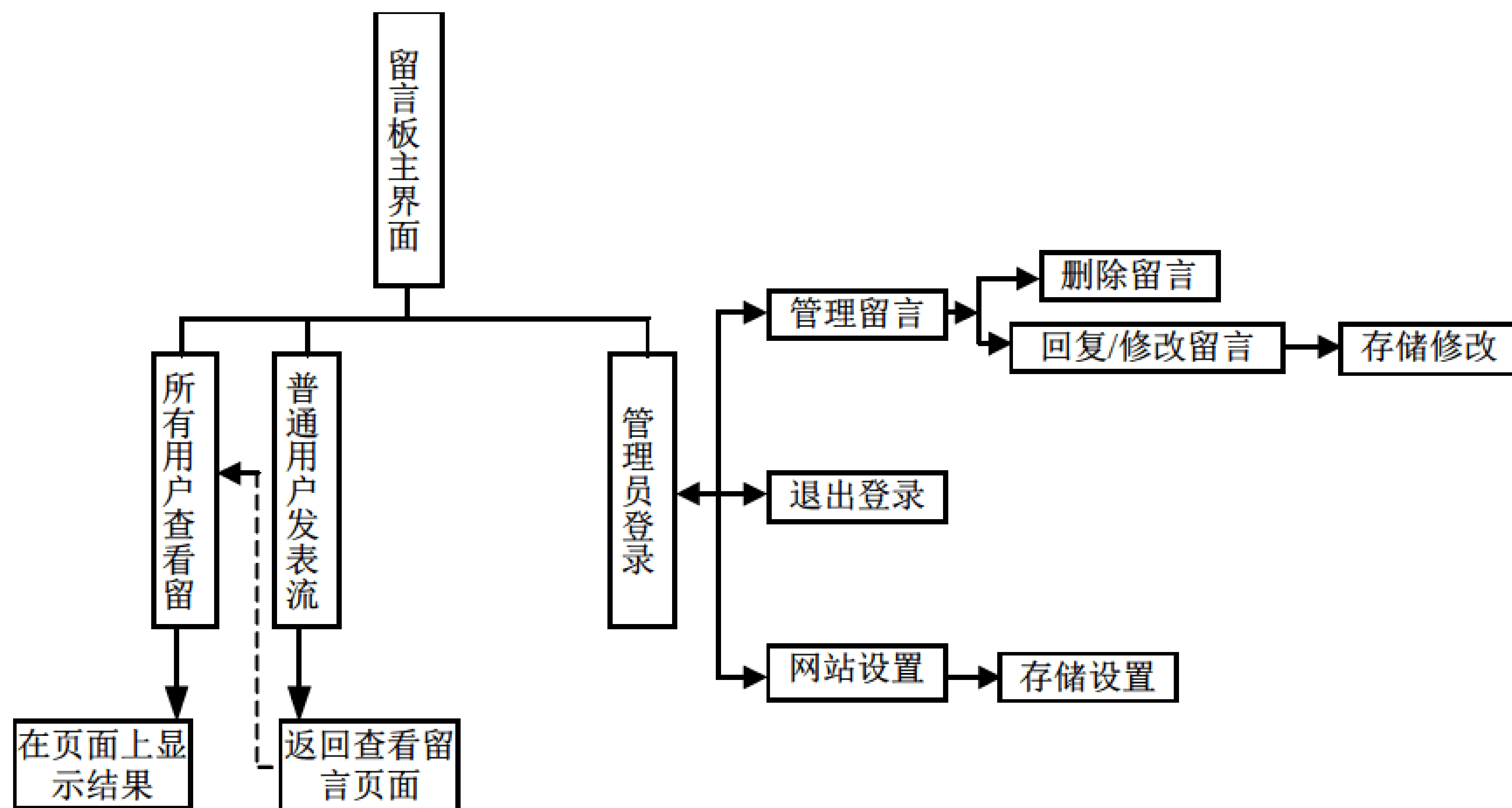


图 13-29　留言板模块的结构

1．页面文件设计

由留言板模块的结构可以看出，该模板主要由以下几个页面文件组成。

- Index.asp 页面：该页面为留言板查看页面，也是模块登录的默认页面，如图 13-30 所示。
- Book_write.asp 页面：该页面为留言发表页面。
- Admin_login.asp 页面：该页面为管理员登录页面，用户可以在页面中输入用户名和密码，如图 13-31 所示。



图 13-30　Index.asp 页面

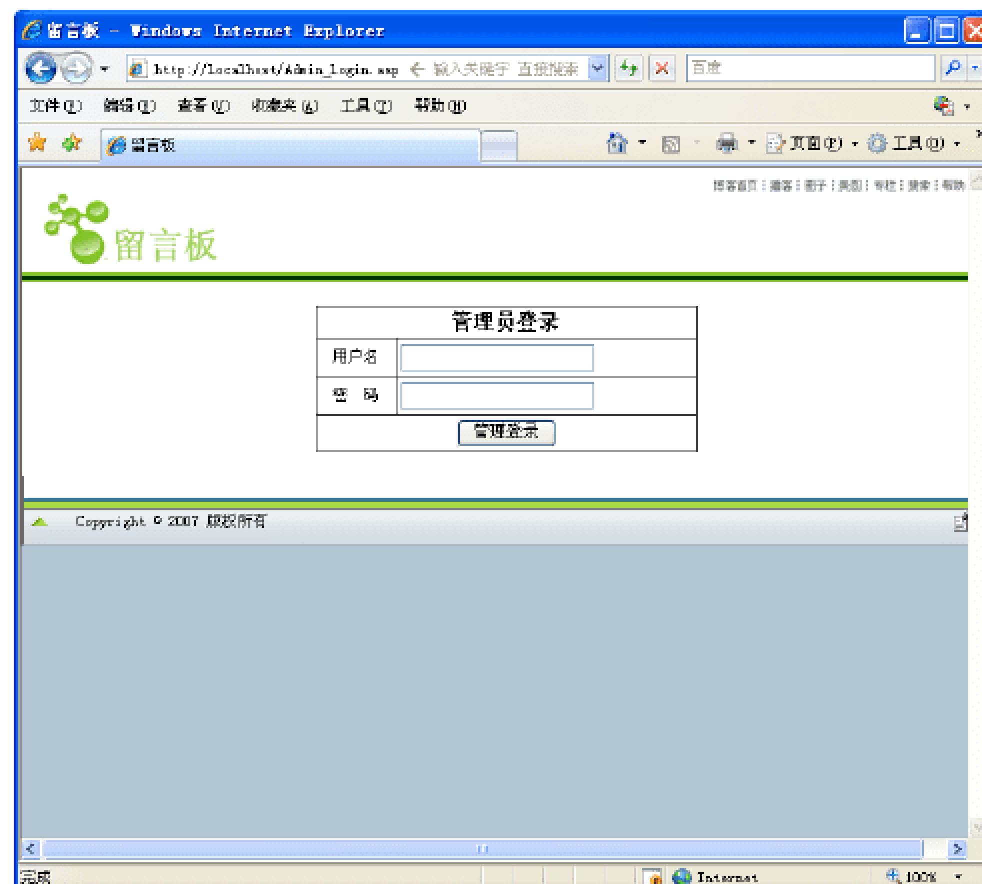


图 13-31　Admin_login.asp 页面

- Admin_main.asp 页面：该页面为留言板管理页面，留言板管理员可以在页面中删除、修改和回复留言，如图 13-32 所示。
- Admin_set.asp 文件：该文件为留言板模块参数设置页面，可以设置留言板网站名称、留言主题最大长度、留言内容最大长度、每页显示的留言条数等，如图 13-33 所示。



图 13-32　Admin_main.asp 页面

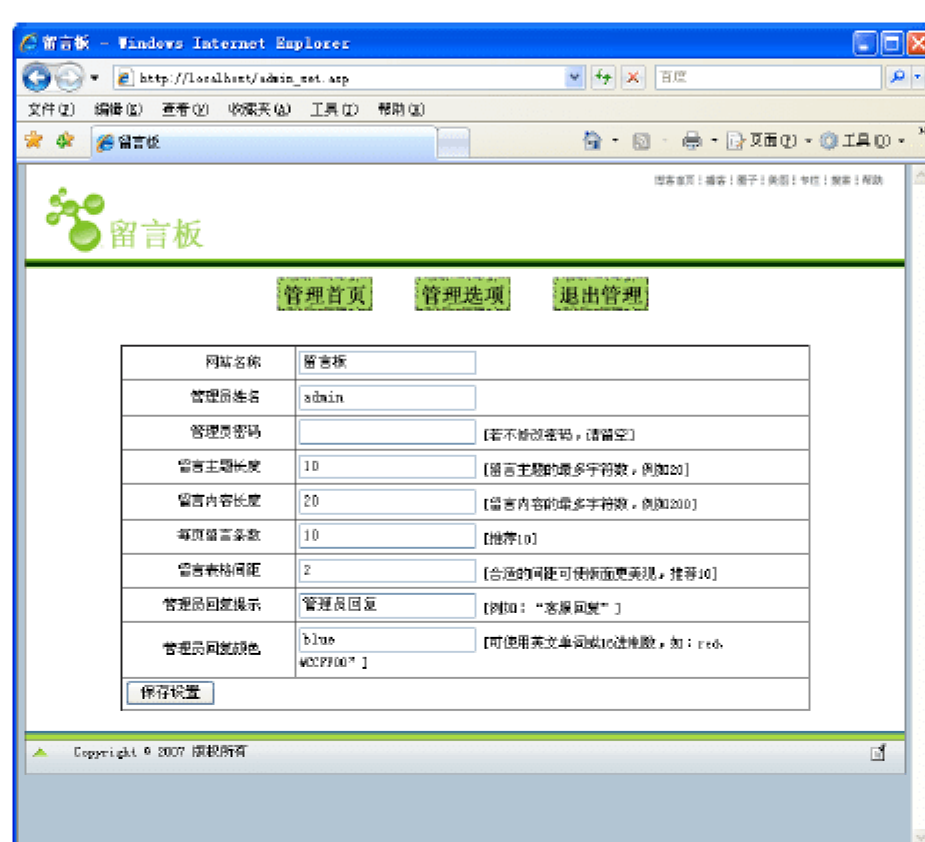


图 13-33　Admin_set.asp 页面

- Admin_logout.asp 文件：该文件为管理员退出登录文件。
- Conn.asp 文件：该文件为数据库连接文件。

2. 数据库设计

本章实例所创建的网站留言板模块需要使用如表 13-2 所示的 Web 数据表保存留言板设置数据，使用表 13-3 所示的 Note_Admin 数据表保存系统管理员的信息数据，使用表 13-4 所示的 Note_Comment 数据表保存留言板的留言信息。

Web 数据表包含网站名称、留言主题最大长度、留言内容最大长度、留言用户是否必须填写电子邮箱、留言页面每页显示留言的条数和显示留言的表格间距等数据。

表 13-2　Web 数据表的结构

字　段	字段类型	主关键字	说　明
ID	自动编号	*	编号
SiteName	文本	—	网站名称
SubjectMaxLength	数字	—	留言主题最大长度
MaxLength	数字	—	留言内容最大长度
Mail	数字	—	留言用户是否必填电子邮箱
Distance	数字	—	显示留言的表格间距
Counts	数字	—	每页显示的留言条数

Note_Admin 数据表包含系统管理的名称、登录密码以及回复留言的颜色和回复留言提示等数据。

表 13-3　Note_Admin 数据表的结构

字　段	字段类型	主关键字	说　明
ID	自动编号	*	编号
Admin	文本	—	用户名
Pwd	文本	—	密码
ReplyColor	文本	—	回复留言颜色
IsReply	文本	—	回复留言提示

Note_Comment 数据表包含留言用户姓名、电子邮箱、主页、IP 地址、其他联系方式以及留言主题、内容、提交时间等数据。

表 13-4　Note_Comment 数据表的结构

字　段	字段类型	主关键字	说　明
ID	自动编号	*	编号
Name	文本	—	留言用户的姓名
Email	文本	—	留言用户的电子邮箱

(续表)

字　段	字 段 类 型	主 关 键 字	说　明
URL	文本	—	留言用户的主页
IP	文本	—	留言用户的 IP 地址
Communication	文本	—	留言用户的其他联系方式
Subject	文本	—	留言主题
Comment	文本	—	留言内容
PostDate	日期时间	—	提交留言的时间
Reply	文本	—	留言的回复内容
ReplyDate	日期时间	—	回复留言的时间
ReplyName	文本	—	回复留言的管理员名称

13.2.3　实例操作

下面将结合本章第 13.2.2 节所介绍的留言板模块结构，通过实例介绍创建一个网站留言板的方法。用户在进行实例操作之前，应结合本书第 1.2 节和第 1.3 节所介绍的方法在本地计算机上构建一个 ASP 应用程序的开发环境。

1. 创建数据库

下面将根据表 13-2 所示的数据库结构，为留言板模块创建数据库。

【练习 13-13】根据表 13-2～表 13-4 所示的内容创建数据表 Web、Note_Admin 和 Note_Comment。

(1) 选择“开始”|“程序”|Microsoft Office| Access 2003 命令，启动 Access 2003 数据库，然后选择“文件”|“新建”命令，打开“新建文件”对话框。

(2) 单击“空数据库”按钮，在打开的“文件新建数据库”对话框的“文件名”文本框中输入 NOTES.mdb。

(3) 单击“创建”按钮，在打开的 Access 2003 的工作界面的工作区域中选择“对象”列表框中的“表”选项，然后在对话框右侧的选项区域中双击“使用设计器创建表”选项打开数据表的设计检视窗口。

(4) 根据表 13-2 所示的内容，创建数据表 Web，完成后的效果如图 13-34 所示。

(5) 根据表 13-3 所示的内容，创建数据表 Note_Admin，完成后的效果如图 13-35 所示。

(6) 根据表 13-4 所示的内容，创建数据表 Note_Comment，完成后的效果如图 13-36 所示。

(7) 完成以上操作后 NOTES 数据库如图 13-37 所示。这时，用户可以关闭 Access 2003 数据库。

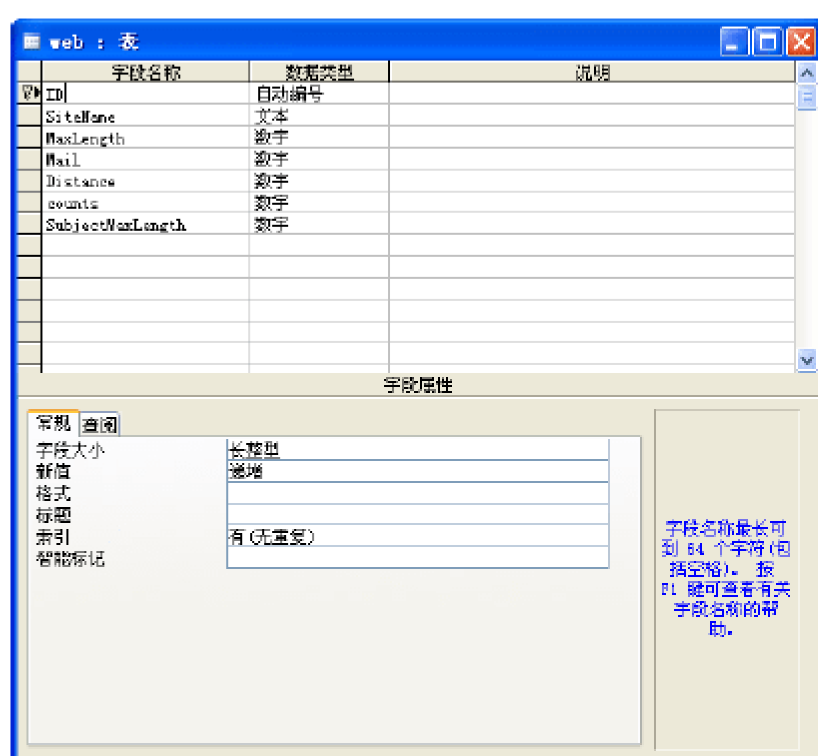


图 13-34　创建数据表 Web

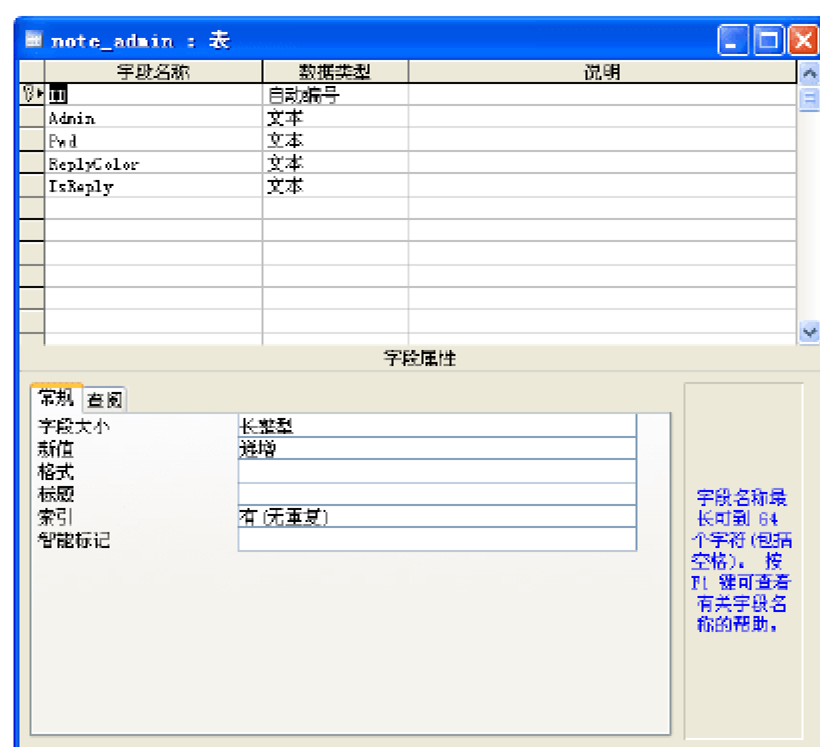


图 13-35　创建数据表 Note_Admin

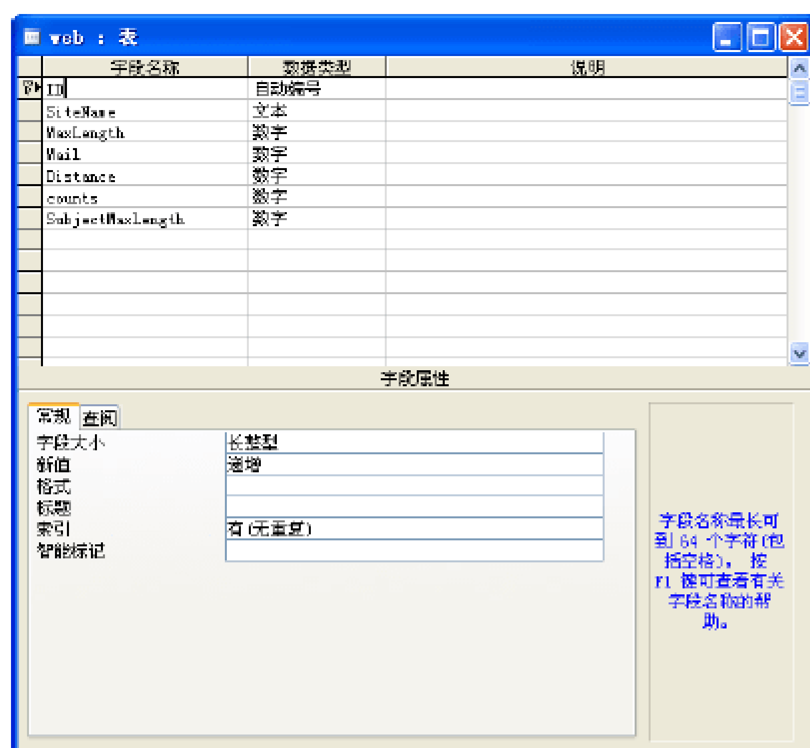


图 13-36　创建数据表 Note_Comment

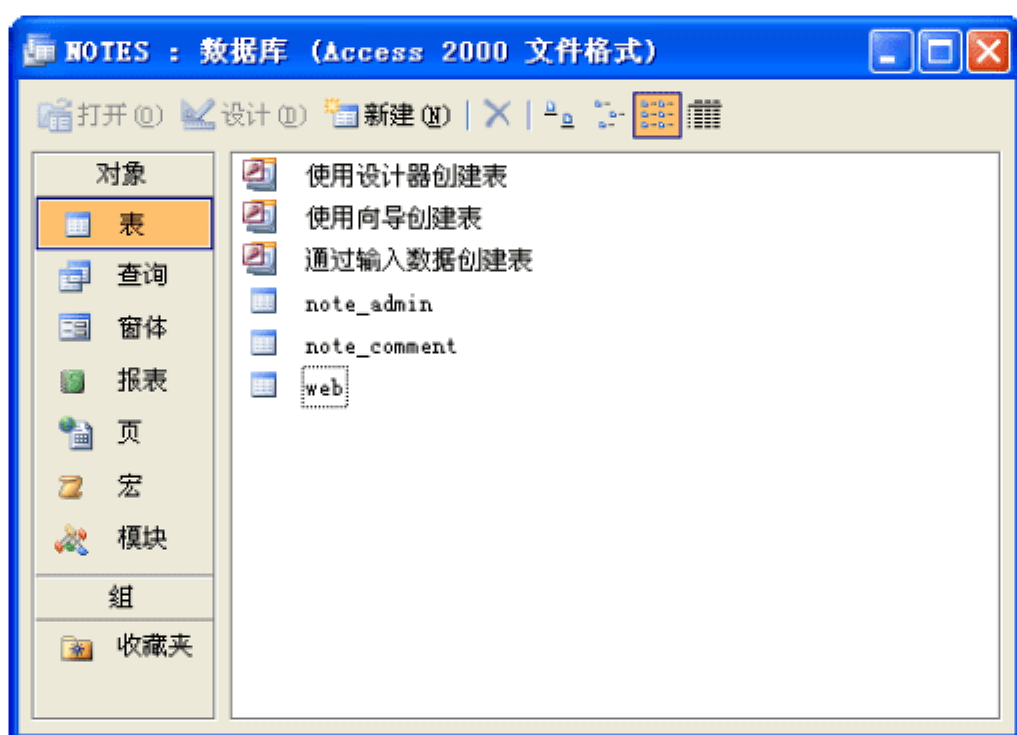


图 13-37　NOTES 数据库

2. 创建数据库连接文件

完成数据库的创建工作后，用户可以参考下面练习所介绍的方法，创建 ASP 网页数据库连接文件 Conn.asp。

【练习 13-14】创建数据库连接文件 Conn.asp，连接数据库 NOTES。

(1) 创建数据库链接文件 Conn.asp，其代码如下所示：

```
<%
Dim database,conn,connstr
Response.Buffer=True
database = "notes.mdb"
Set conn=server.createobject("ADODB.Connection")
connstr="provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.MapPath(database)
conn.Open connstr
Set WBrs = Server.CreateObject("ADODB.RecordSet")
WBrs.Open "Select * from web",conn,1,3
sitename = WBrs("SiteName")
maxlength = WBrs("MaxLength")
```

```
mail = WBrs("Mail")
distance = WBrs("Distance")
counts = WBrs("Counts")
subjectmaxlength = WBrs("SubjectMaxLength")
WBrs.Close
Set WBrs = nothing
%>
```

(2) 将 Conn.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

3. 创建留言查看页面

留言查看页面(Index.asp)是用户进入留言板模块后首先打开的页面。在该页面中，将显示留言者的姓名、IP 地址，留言信息的主题、内容和管理员回复等信息，如图 13-38 所示。

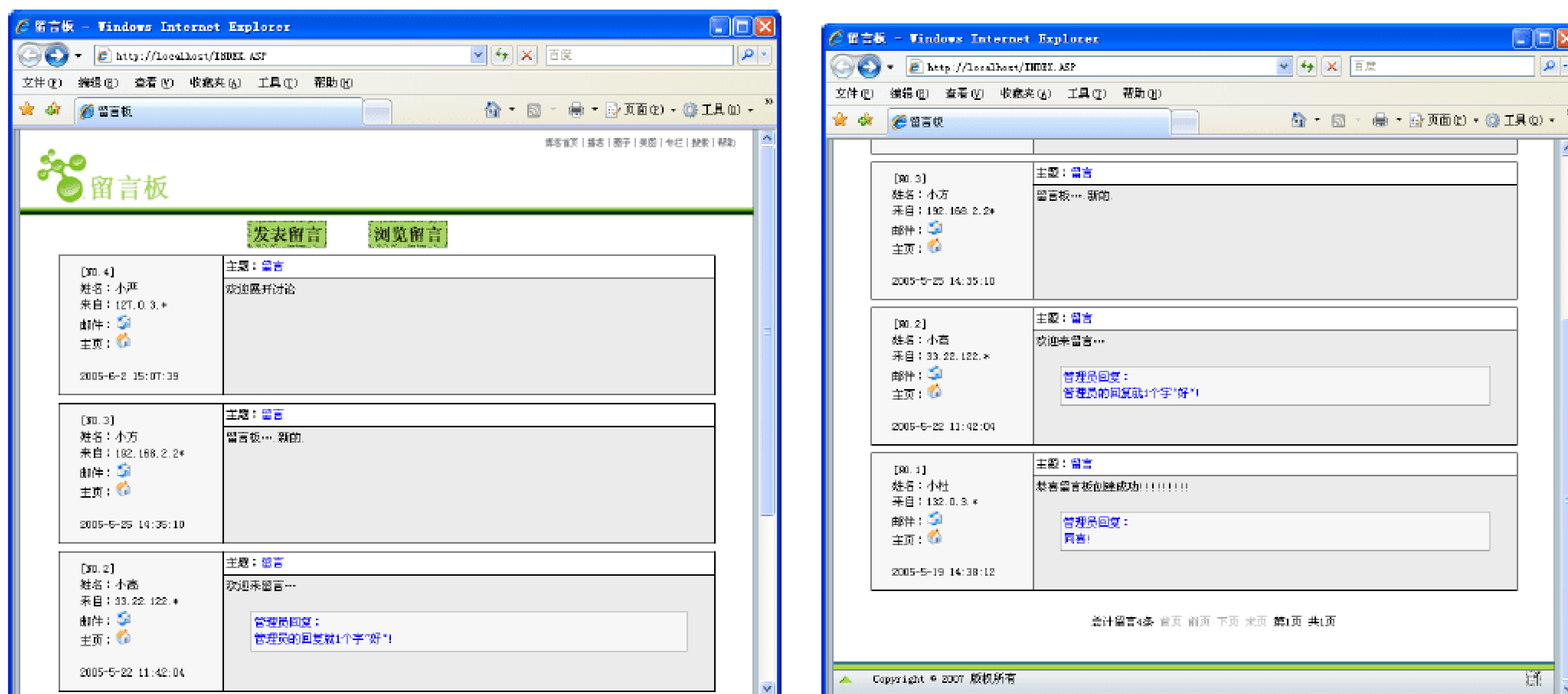


图 13-38　留言查看页面

【练习 13-15】创建网站留言板留言查看页面。

(1) 在 C:\Inetpub\wwwroot 文件夹中创建一个名称 IMAGES 的子文件夹，然后将本节实例所需的图片文件复制至该文件夹中。

(2) 创建留言查看页面 Index.asp，其代码如下所示：

```
<!--#include file="conn.asp"-->
<HTML><HEAD>
<TITLE><%=sitename%></TITLE>
<link rel="stylesheet" href="book.css" type="text/css">
</HEAD>
<body background=images/bg.gif topmargin="0" leftmargin="0">
<table width="760" border=0 cellspacing=0 cellpadding=0 bgcolor="#FFFFFF" align=center
        class="grayline">
<tr><td><img border="0" src="IMAGES/TOPS.gif" width=758></td></tr>
```

```
</table>
<center>
<table width="760" border=0 cellspacing=0 cellpadding=0 align=center bgcolor="#FFFFFF"
          class="grayline">
 <tr>
 <td align=center height=25 colspan="2" width="680">
 <a href=book_write.asp><img border=0 src=images/write.gif title="我要写留言"></a>
          
 <a href=index.asp><img border=0 src=images/read.gif title="留言本首页"></a>
 </td>
 </tr>
 <tr>
 <td>
 <%
 set rs=Server.CreateObject("ADODB.RecordSet")  '创建 RecordSet 对象
 rs.open "select * from note_comment order by id desc",conn,1,3
'从数据库中读取留言，并保存在 rs 中
 if rs.eof and rs.bof then                          '不存在留言记录
 response.redirect "book_write.asp"                 '转到撰写留言页面
 else                                               '存在记录
        if counts=0 or counts="" then counts=10      '每页留言条数
        rs.pageSize = counts                         '每页记录数
        allpages = rs.pageCount                      '总页数
        page = Request("page")                       '从浏览器取得当前页
        If not isNumeric(page) then page=1           'page 不是数字
        if isEmpty(page) or Cint(page) < 1 then      'page 为空或小于 1
        page = 1
        elseif Cint(page) >= allPages then           'page 值大于总页数
        page = allPages
        end if
        rs.AbsolutePage = page                       '转到某页头部
        Do While Not rs.eof
               UserName=rs("Name")                   '用户名
               Comments=rs("Comment")                '内容
               Reply=rs("Reply")                     '回复
               Usermail=rs("Email")                  '邮件
               url=rs("Url")                         '主页
               replyname = rs("ReplyName")           '回复人姓名
               I=I+1                                 '序号
               '留言序号
               temp=RS.RecordCount-(page-1)*rs.pageSize-I+1
 %>
<table cellSpacing="1" cellPadding="3" width="680" align="center" bgColor="#000000"
```

```
     border="0">
  <tr>
<td vAlign="top" width="25%" bgColor="#f7f7f7" rowSpan="2" align=center>
<table border=0 width=80%><tr><td align=center></td></tr>
<tr><td>[NO.<%=temp%>]<br>
姓名：<%=UserName%><br>
来自：<%=left(rs("ip"),(len(rs("ip"))-1))+"*"%><br>
邮件：<a href=mailto:<%=Usermail%>><img src=images/mail.gif border=0></a><br>
主页：<a href="<%=URL%>" target='_blank'><img src=images/home.gif border=0></a><br><br>
<%=rs("Postdate")%></td></tr>
</table>
</td>
<td width="75%" height="20" bgColor="#ffffff">主题：<font
     color=blue><%=rs("subject")%></font> </td>
  </tr>
  <tr>
<td vAlign="top" bgColor="#ebebeb" width='75%' height=120 onMouseOver="bgColor='#FFffff'"
     onMouseOut="bgColor='#ebebeb'">
<%=Comments%>
<br><br>
<%if rs("Reply")<>"" then
Set RCrs = Server.CreateObject("ADODB.RecordSet")
RCrs.Open "Select * from note_Admin where Admin="&replyname,conn,1,3
%>
<table cellSpacing="1" cellPadding="3" width="90%" align="center" bgColor="darkgray"
       border="0">
   <tr><td vAlign="top" bgColor="#f7f7f7">
<font color=<%=RCrs("replycolor")%>><%=RCrs("isreply")%>：<br><%=Reply%></font>
</td></tr></table><br>
<%
RCrs.Close
Set RCrs = nothing
end if%>
</td>
  </tr>
</table>
<table cellSpacing="0" cellPadding="0" width="680" align="center" bgColor="#FFFFFF"
      border="0" height = "<%=distance%>">
<TR><TD height=8> </TD></TR></TABLE>
<%
           counts = counts-1                      '每页显示条数减 1
           rs.movenext                            '指向下一条记录
           if rs.eof then exit do                 '已经到最后一条记录则跳出
```

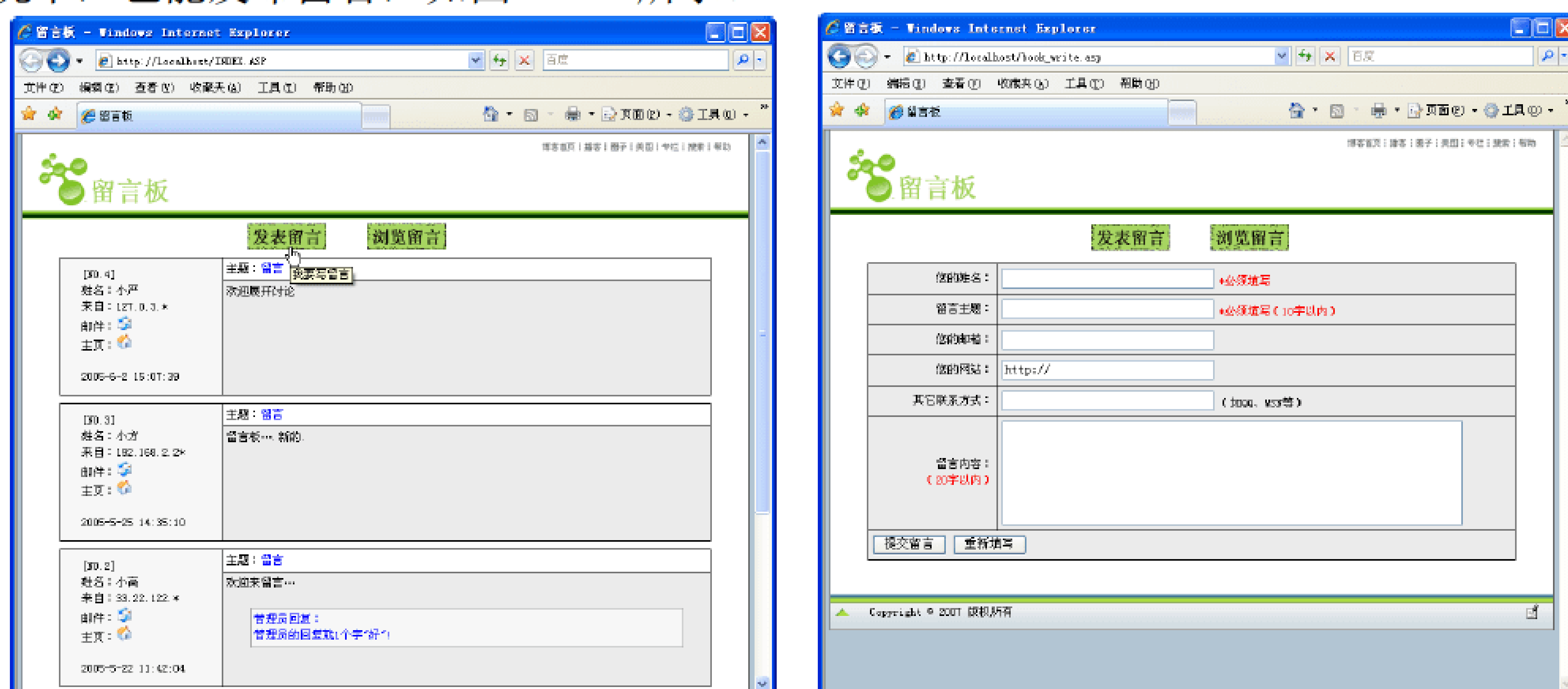
```
 loop                                    '结束循环
 'End If
 %>
</td></tr>
<tr><td height=50 valign=top align = center>
<%
'显示总留言条数
response.write "<br>       总计留言
        "&RS.RecordCount&"条  "
if page = 1 then                                             '到达第一页
 response.write "<font color=darkgray>首页  前页</font>"
else                                                         '不是第一页
 'request.ServerVariable("script_name")获得程序的虚拟路径
 response.write "<a href="&request.ServerVariables("script_name")&"?page=1>首页</a><a
      href="&request.ServerVariables("script_name")&"?page="&page-1&">前页</a>"
end if
if page = allpages then                                      '到达最后一页
 response.write "<font color=darkgray>  下页  末页</font>"
else                                                         '不是最后一页
 '建立超级链接
 response.write " <a href="&request.ServerVariables("script_name")&"?page="&page+1&">下页
    </a> <a href="&request.ServerVariables("script_name")&"?page="&allpages&">末页</a>"
end if
'显示留言当前的页数信息
response.write "  第"&page&"页  共"&allpages&"页  "
%>
</td></tr>
</table>
<table width="760" border=0 cellspacing=0 cellpadding=0 bgcolor="#FFFFFF" align=center
        class="grayline">
<tr><td>   </td></tr>
<tr><td width=30 height=37 background="IMAGES/down.gif"><a href=#top><img
        src=images/up.gif border=0></a></td>
<td height=37 background="IMAGES/down.gif">  Copyright &copy; 2007   版权所有
   </td>
<td width=30 height=37 background="IMAGES/down.gif"><a href=admin_login.asp title="管理留
          言本"><img src=images/admin.gif border=0></a></td>
</tr>
</table>
</center>
</body>
</html>
<%
```

```
End If
conn.CLose                                                    '关闭数据库
Set conn = nothing
%>
```

(3) 选择“文件”|“另存为”命令，将上面输入的代码以文件名 Index.asp 保存至 C:\Inetpub\wwwroot 文件夹中。

4. 创建留言发表页面

本节实例所介绍的留言发表页面 book_write.asp 允许留言发布者在无需登录留言板的情况下，也能发布留言，如图 13-39 所示。



单击“发表留言”按钮　　留言发表页面

图 13-39　打开留言发表页面

【练习 13-16】创建留言板模块的留言发表页面。

(1) 创建留言发表页面 book_write.asp，其代码如下所示：

```
<!--#include file="conn.asp"-->
<%
if request("send")="ok" then                    '用户提交留言
 username=trim(request.form("username"))
'读取表单传递过来的姓名数据，Trim 函数去掉字符串的前导和后续空格
 usermail=trim(request.form("usermail"))        '读取表单传递过来的用户邮箱数据
 '没有填写用户姓名或留言主题或留言内容
 if username="" or request.form("Comments")="" or request.form("subject") = "" then
 '给出提示，并返回到前一页
 response.write "<script>alert('填写资料不完整，请检查后重新输入！');history.back();</script>"
 response.end
 end if
 if mailyes=0 then                              '邮箱为必填时检查邮箱是否合法
 '邮箱地址中存在字符“.”或“@”，或邮箱长度小于 10 或大于 50 都是非法邮箱地址
```

```
        if Instr(usermail,".")<=0 or Instr(usermail,"@")<=0 or len(usermail)<10 or len(usermail)>50
         then
 '给出提示并返回到前一页
 response.write "<script>alert('您输入的电子邮件地址格式不正确，请检查后重新输入！
    ');history.back();</script>"
        response.end
        end if
 end if
 '留言主题长度大于要求的长度
 if len(request.form("subject"))>subjectmaxlength then
 '给出提示并返回到前一页
 response.write "<script>alert('留言主题太长了，请不要超过"&subjectmaxlength&"个字符！
    ');history.back();</script>"
 response.end
 end if
'留言内容长度大于要求的长度
 if len(request.form("Comments"))>maxlength then
 '给出提示并返回到前一页
 response.write "<script>alert('留言内容太长了，请不要超过"&maxlength&"个字符！
     ');history.back();</script>"
 response.end
 end if
 set rs=Server.CreateObject("ADODB.RecordSet")          '将留言保存到数据库中
 sql="select * from note_comment"                       '从数据库中查询留言记录
 rs.open sql,conn,1,3                                   '执行查询操作，并将结果保存在 rs 中
 rs.Addnew                                              '添加一条记录
 rs("Name")=Request.Form("username")                    '姓名
 rs("Comment")=Request.Form("comments")                 '留言内容
 rs("Email")=Request.Form("usermail")                   '邮箱地址
 rs("url")=Request.Form("url")                          '主页
 rs("Communication")=Request.Form("communication")      '其他联系方式
 rs("IP")=Request.serverVariables("REMOTE_ADDR")        'IP 地址
 rs("PostDate") = now()                                 '留言提交时间
 rs("Subject") = Request.Form("subject")                '留言主题
 rs.Update                                              '更新数据库
 rs.close                                               '关闭 RecordSet 对象
 set rs=nothing
 '提交成功，给出提示，并返回到留言查看界面
 response.write "<script>alert('留言提交成功，单击“确定”返回留言列表！
    ');location.href='index.asp';</script>"
 response.end
end if
%>
```

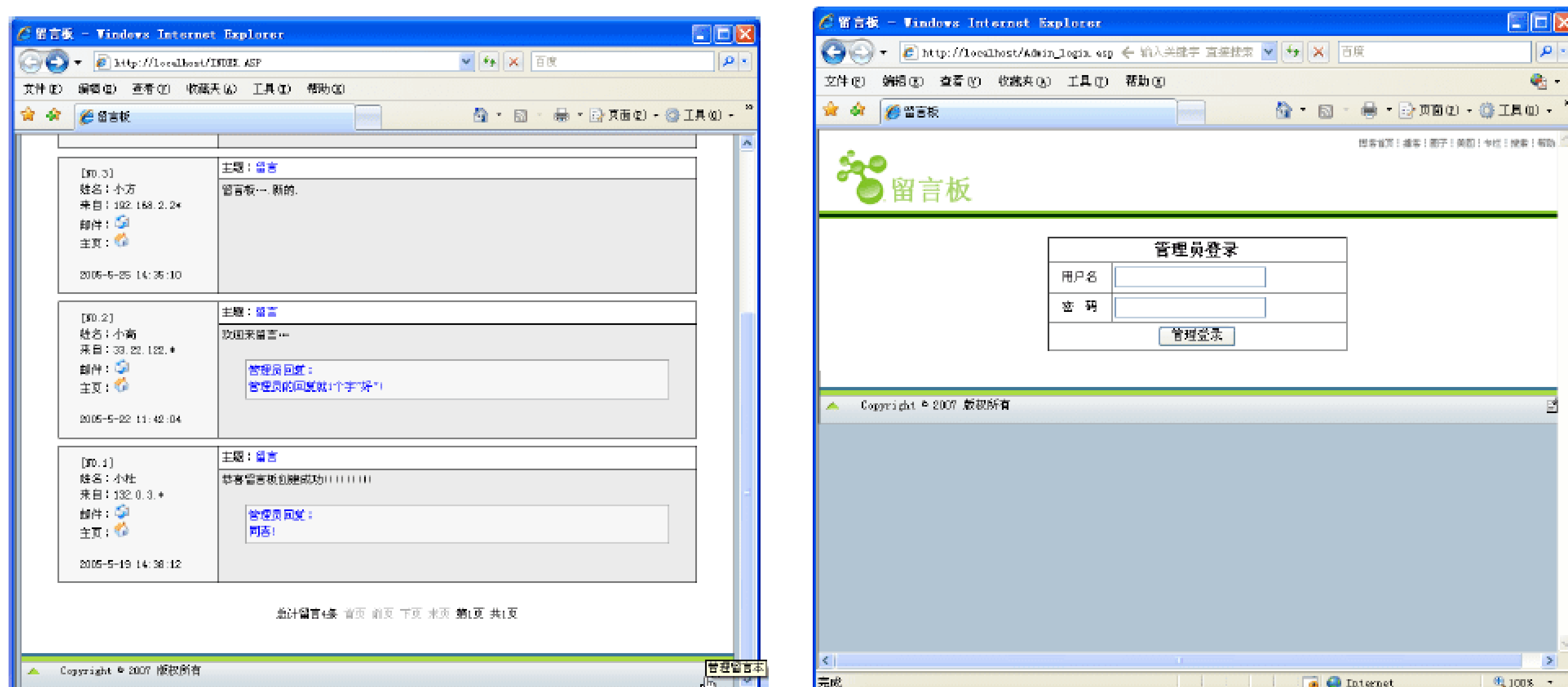
```
<HTML><HEAD>
<TITLE><%=sitename%></TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta name="description" content="<%=sitename%>">
<meta name="keywords" content="<%=sitename%>">
<link rel="stylesheet" href="book.css" type="text/css">
</HEAD>
<body background=images/bg.gif topmargin="0" leftmargin="0">
<table width="760" border=0 cellspacing=0 cellpadding=0 bgcolor="#FFFFFF" align=center
     class="grayline">
<tr><td><img border="0" src="IMAGES/TOPS.gif" width=758></td></tr>
</table>
<center>
<table width="760" border=0 cellspacing=0 cellpadding=0 align=center bgcolor="#FFFFFF"
    class="grayline">
<tr><td align=center height=50>
<img border=0 src=images/write.gif>        <a href=index.asp><img
    border=0 src=images/read.gif title="我要看留言"></a>
</td></tr>
<tr><td>
<form action=book_write.asp method=post name="book">
<table cellSpacing="1" borderColorDark="#ffffff" cellPadding="4" width="680" align="center"
    bgColor="#000000" borderColorLight="#000000" border="0">
 <tr bgColor="#ebebeb">
 <td    width="20%" align=right>您的姓名：</td>
 <td ><input type=text name="UserName" size="30" maxlength=16>
 <font color="#FF0000">*必须填写</font></td>
 </tr>
 <tr bgColor="#ebebeb">
 <td    width="20%" align=right>留言主题：</td>
  <td ><input type=text name="subject" size="30"   maxlength=50>
  <font color="#FF0000">*必须填写(<%=subjectmaxlength%>字以内)</font></td>
  </tr>
  <tr bgColor="#ebebeb">
  <td    width="20%" align=right>您的邮箱：</td>
  <td ><input type=text name="UserMail" size="30"   maxlength=50>
  <%if mail=0 then%><font color="#FF0000">*必须填写</font><%end if%></td>
  </tr>
  <tr bgColor="#ebebeb">
  <td    width="20%" align=right>您的网站：</td>
  <td><input type=text value="http://" name="url" size="30"   maxlength=100></td>
  </tr>
  <tr bgColor="#ebebeb">
```

```
    <td    width="20%" align=right>其它联系方式：</td>
    <td><input type=text value="" name="communication" size="30"   maxlength=100> (如 QQ、
         MSN 等)</td>
  </tr>
  <tr bgColor="#ebebeb">
  <td    width="20%" align=right>留言内容：<br><font color=red>(<%=maxlength%>字以
         内)</font></td>
   <td><textarea name="Comments" rows="7" cols="66" style="overflow:auto;"></textarea></td>
   <tr bgColor="#ebebeb">
   <td colSpan="2"><input type="submit" value="提交留言" name="Submit">
   <input type="reset" value="重新填写" name="Submit2"><input type=hidden name=send
         value=ok></td>
</tr>
</table>
</form>
</table>
<table width="760" border=0 cellspacing=0 cellpadding=0 bgcolor="#FFFFFF" align=center
         class="grayline">
<tr><td>   </td></tr>
<tr><td width=30 height=37 background="IMAGES/down.gif"><a href=#top><img
         src=images/up.gif border=0></a></td>
<td height=37 background="IMAGES/down.gif">  Copyright &copy; 2007   版权所有
         </td>
<td width=30 height=37 background="IMAGES/down.gif"><a href=admin_login.asp title="管理留
         言本"><img src=images/admin.gif border=0></a></td>
</tr>
</table>
</center>
</body>
</html>
<%
conn.CLose
Set conn = nothing
%>
```

(2) 将 book_write.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

5. 创建管理员登录页面

网站留言板模块的管理员登录页面 admin_login.asp 允许留言板管理员登录留言板管理界面，如图 13-40 所示。



单击“管理留言本”按钮　　　　管理员登录页面

图 13-40　打开管理员登录页面

【练习 13-17】创建留言板模块的管理员登录页面。

(1) 创建留言板模块管理员登录页面 admin_login.asp，其代码如下所示。

```
<!--#include file = "conn.asp"-->
<%
if request.QueryString("act")="login" then
'管理员登录
user = Request.Form("username")
'读取从表单传递过来的用户名数据
pass = Request.Form("password")
'读取从表单传递过来的密码数据
Set NArs=Server.CreateObject("Adodb.RecordSet")
'建立 RecordSet 对象
'从数据库中查找 Admin 值为 user，pwd 值为 pass 的记录
sql="Select * from note_admin where Admin='"&user&"' and pwd='"&pass&"'"
'执行查询操作，并将结果保存在 NArs 中
 NArs.open sql,conn,1,1
 If not(NArs.bof and NArs.eof) Then
'存在满足条件的记录，则登录成功
 Session("login") = "ok"
'将 Session 变量 login 置为 ok
 Session("user") = NArs("Admin")
'将当前登录管理员的用户名赋给 Session 变量 user
 Session("id") = NArs("id")
'将当前登录管理员的 id 赋给 Session 变量 id
 Response.Redirect "admin_main.asp?action=main"
 '页面转到留言管理界面
Else
```

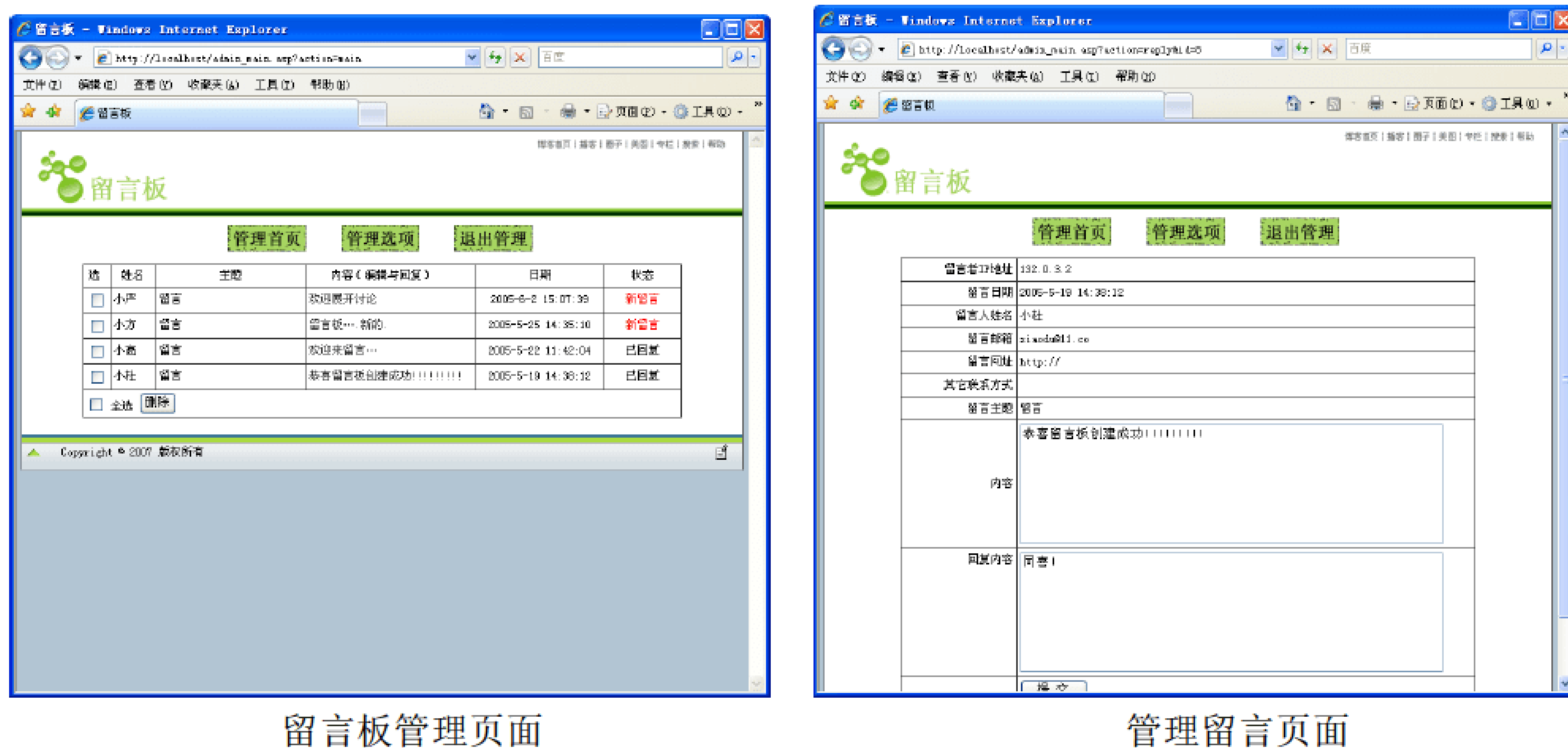
```
'不存在满足条件的记录，给出提示，返回到前一页
 Response.Write("<script>alert('管理员用户名称或密码输入有误，请检查后重新输入！
        ');history.back()</script>")
    Response.End
    End If
    NArs.Close
'关闭 RecordSet 对象
    Set NArs= nothing
End If
%>
<HTML><HEAD>
<TITLE><%=sitename%></TITLE>
<title>管理员登录</title>
<link rel="stylesheet" href="book.css" type="text/css">
<script language="VBScript">
function check()                         '检查是否输入用户名、密码
 If booklogin.username.value="" Then     '“用户名”文本框的值为空
        alert("用户名不能为空!")           '给出提示
        booklogin.username.focus()       '将“用户名”文本框置为输入焦点
        return false
 End If
 If booklogin.password.value="" Then     '“密码”文本框的值为空
        alert("密码不能为空!")             '给出提示
        booklogin.password.focus()       '将“密码”文本框置为输入的焦点
        return false
 End If
 return true
End Function
</script>
</HEAD>
<%
if session("login")<>"ok" then
%>
<body background=images/bg.gif topmargin="0" leftmargin="0">
<table width="760" border=0 cellspacing=0 cellpadding=0 bgcolor="#FFFFFF" align=center
    class="grayline">
<tr><td><img border="0" src="IMAGES/TOPS.gif" width=758></td></tr>
</table>
<center>
 <table width="760" border=0 cellspacing=0 cellpadding=0 align=center bgcolor="#FFFFFF"
     class="grayline">
 <tr><td>
<br>
```

```
<form action="admin_login.asp?act=login" method=post name=booklogin onsubmit = "return
    check()">
<table width="300" border="1" cellpadding="3" bordercolor="#333333" style="border-collapse:
    collapse;" align=center>
<tr>
 <th colspan = 2>管理员登录</th>
<tr><td align=center>用户名</td><td><input type="text" name="username" value="" title="请填写
     用户名"></td></tr>
<tr><td align=center>密  码</td><td><input type="password" name="password"
      value="" title="请填写登录密码"></td></tr>
<tr><td colspan=2 align=center><input type="submit" value="管理登录"><input type="hidden"
      name="login" value="check"></td></tr>
</table>
</form>
<table width="760" border=0 cellspacing=0 cellpadding=0 bgcolor="#FFFFFF" align=center
      class="grayline">
<tr><td>   </td></tr>
<tr><td width=30 height=37 background="IMAGES/down.gif"><a href=#top><img
      src=images/up.gif border=0></a></td>
<td height=37 background="IMAGES/down.gif">  Copyright &copy; 2007   版权所有
       </td>
<td width=30 height=37 background="IMAGES/down.gif"><a href=admin_login.asp title="管理留
       言本"><img src=images/admin.gif border=0></a></td>
</tr>
</table>
</td></tr>
</table>
</center>
</body>
<%
End If
%>
</html>
```

(2) 将 admin_login.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

6. 创建留言板管理页面

留言板管理页面 admin_main.asp 是留言板管理员在 admin_login.asp 页面中成功登录后所进入的页面。该页面不仅包含一个分页的用户留言列表，以每页 10 条的数量显示留言板系统中的留言，而且还具备多个管理员操作按钮，可以允许留言板管理员进行相应的管理操作，如图 13-41 所示。



留言板管理页面　　　　管理留言页面

图 13-41　管理留言板

【练习 13-18】创建留言板模板的留言板管理页面。

(1) 创建留言板管理页面文件 admin_main.asp，其代码如下所示：

```
<!--#include file = "conn.asp"-->
<%
'function lleft(content,lef)
 for le=1 to len(content)
        if asc(mid(content,le,1))<0 then
        lef=lef-2
        else
        lef=lef-1
        end if
        if lef<=0 then exit for
 next
 lleft=left(content,le)
'end function
%>
<HTML>
 <HEAD>
  <TITLE><%=sitename%></TITLE>
  <link rel="stylesheet" href="book.css" type="text/css">
  <script language="JavaScript">
 function CheckAll(form)
 {
        for (var i=0;i<form.elements.length;i++)
        {
         var e = form.elements[i];
         if (e.name != 'chkall')        e.checked = form.chkall.checked;
```

```
        }
        }
    </script>
     </HEAD>
   <body background=images/bg.gif topmargin="0" leftmargin="0">
   <center>
    <table width="760" border=0 cellspacing=0 cellpadding=0 bgcolor="#FFFFFF" align=center
          class="grayline">
     <tr><td><img border="0" src="IMAGES/TOPS.gif" width=758></td></tr>
    </table>
    </center>
    <table width="760" border=0 cellspacing=0 cellpadding=0 align=center bgcolor="#FFFFFF"
          class="grayline">
  <tr><td align=center height=50>
      <a href=admin_main.asp><img border=0 src=images/admin_home.gif title="管理首页"></a>
           
      <a href=admin_set.asp><img border=0 src=images/admin_setup.gif title="管理选项设置"></a>
            
      <a href=admin_logout.asp><img border=0 src=images/loginout.gif title="管理结束后，一定要
        点击这里"></a>
     </td></tr>
     <%
     action=request.QueryString("action")
     if action="main" or action = "" then                          '管理首页
     %>
     <tr><td align=center>
     <table width="630" border="1" cellpadding="3" bordercolor="#333333" style="border-collapse:
          collapse">
    <form name=bookform action=admin_main.asp method=post>
      <tr>
       <td align=center width=18>选</td>
       <td align=center width=41 height=15>姓名</td>
       <td align=center width=150>主题</td>
       <td align=center width=155>内容(编辑与回复)</td>
       <td align=center width=128>日期</td>
       <td align=center width=74>状态</td>
      </tr>
        <%
       dim rs,msg_per_page                                     '定义变量
       dim sql
       msg_per_page = 10                                      '每页显示记录数
       Set rs=Server.CreateObject("ADODB.Recordset")          '创建 RecordSet 对象
       sql="select * from note_comment order by id desc"
```

```
 '从数据库中查询所有记录并按照 id 降序排列
rs.pagesize = msg_per_page                          '每页显示记录数
rs.open sql,conn,1,3                                '执行查询操作

if rs.eof and rs.bof then                           '数据库中不存在记录
    response.write "<tr><td colspan=6 align=center><BR>暂时没有留言
      <BR><BR></td></tr>"
end if

if not (rs.eof and rs.bof) then                     '检测记录集是否为空
    totalrec = RS.RecordCount                       '总记录条数
    if rs.recordcount mod msg_per_page = 0 then
    '计算总页数，recordcount 为数据的总记录数
    n = rs.recordcount\msg_per_page                 'n 为总页数
    else
    n = rs.recordcount\msg_per_page+1               'n 为总页数
    end if
    currentpage = request("page")                   'currentpage 为当前页
    If currentpage <> "" then                       'currentpage 值存在
    currentpage = cint(currentpage)                 'cint 函数将 currentpage 转换为整数
    if currentpage < 1 then                         'currentpage 值小于 1
    currentpage = 1
    end if
    if err.number <> 0 then                         '产生某种错误
    err.clear                                       '清除错误信息
    currentpage = 1
    end if
    else                                            'currentpage 值为空
    currentpage = 1
    End if
    '当前页大于总页数
    if currentpage*msg_per_page > totalrec and not((currentpage-1)*msg_per_page <
      totalrec)then
    currentPage=1
    end if
    rs.absolutepage = currentpage               'absolutepage 设置指针指向某页开头
    rowcount = rs.pagesize                      'pagesize 设置每一页的数据记录数
    dim i
    dim k
    Do while not rs.eof and rowcount>0          '循环显示所有的记录
        content=rs("Comment")                   '留言内容
        subject = rs("Subject")                 '留言主题
        reply=rs("reply")                       '回复内容
```

```
                UserName=rs("Name")                '留言者姓名
                '使用 Response 对象的 write 方法在网页上显示复选框、留言者姓名、
                '留言主题、留言内容、留言日期和留言状态
                Response.write "<tr><td align=center><input type='checkbox'
                    value='"&rs("ID")&"' name=id>"
                Response.write "</td><td>"&UserName&"</td><td>"
                'Response.write lleft(server.htmlencode(subject),50)
                Response.write subject
                '建立到管理员回复留言的超级链接
                Response.write "</td><td><a href='admin_main.asp?action=reply&id="&rs("ID")&"'>"
                'Response.write lleft(server.htmlencode(content),50)
                Response.write content
                Response.write "</a></td><td  align=center>"&rs("Postdate")&"</td><td
                    align=center>"
                if Isnull(Reply) then                        '没有留言回复
                response.write "<font color=red>新留言</font>"
                else                                         '管理员已经回复留言
                response.write "已回复"
                end if
                rowcount=rowcount-1        '显示一条记录后，将每页未显示的记录数减 1
                rs.movenext                                  '指向下一条记录
            loop
        end if
        rs.close                                             '关闭 RecordSet 对象
        set rs=nothing
        %>
        <tr><td colspan=6 width="616"><input type="checkbox" name=chkall
          onclick="CheckAll(this.form)"> 全选
            <input type="submit" name="action" value="删除" onclick="{if(confirm('该操作不可恢
              复！\n\n 确实删除选定的留言？')){this.document.Prodlist.submit();return true;}return false;}">
        </td></tr></form></table>
        </td></tr>
        <tr><td>
        <%
        call listPages()                    '调用 listPages 过程显示记录数目信息、页数信息
        %>
        </td></tr>
<%end if
if request("action")="删除" then                          '删除操作
        delid=replace(request.form("id"),"'","")
        '获得要删除记录的 id，Replace 过程将获得的 id 字符串中的"'"用空字符""替换
        call delfeedback()
end if
```

```
 if request("action")="reply" then
        id=request("id")
        call detailfeedback()
 end if
'end if
'分页
sub listPages()
 if n <= 1 then exit sub
%>
共<%=totalrec%>条留言
<%if currentpage = 1 then%>
<font color=darkgray>首页  前页</font>
<%else%>
<a href="<%=request.ServerVariables("script_name")%>?page=1">
首页</font></a> <a href="<%=request.ServerVariables("script_name")%>?page=<%=currentpage-1%>">前页</a>
<%end if%>
<%if currentpage = n then%>
<font color=darkgray >下页  末页</font>
<%else%>
<a href="<%=request.ServerVariables("script_name")%>?page=<%=currentpage+1%>">下页</a>
<a href="<%=request.ServerVariables("script_name")%>?page=<%=n%>">末页</a>
<%end if%>
  第<%=currentpage%>页  共<%=n%>页
<%
end sub
sub delfeedback()                                              '删除留言
 if delid="" or isnull(delid) then                             '要删除的记录 id 为空
        '则给出提示
        response.write "<script>alert('操作失败，没有选择合适参数，请单击“确定”返回！
            ');</script>"
        response.redirect "admin_main.asp"                     '返回到留言管理界面
        response.end
 else                                                          'id 存在
        '执行数据库中的删除操作，删除数据库中 id 值在 delid 数组中的记录
        conn.Execute("delete * from note_comment where ID in ("&delid&")")
        '删除成功，给出提示
        response.write "<script>alert('留言删除成功，请单击“确定”返回！');</script>"
        response.redirect "admin_main.asp"                     '返回到留言管理界面
        response.end
 end if
end sub
sub detailfeedback()                                           '留言回复
```

```
if id="" then                                          '要回复的留言 id 为空
        '给出提示
        response.write "<script>alert('无此留言编号，请单击“确定”返回！');</script>"
        response.redirect "admin_main.asp"             '返回到留言管理界面
        response.end
end if
'修改留言资料
if request("send")="ok" then                           '提交留言回复内容
        set rs=server.createobject("adodb.recordset")  '创建 RecordSet 对象
        sql = "select * from note_comment where ID="&id
        '从数据库中查询 id 值为 id 的记录
        rs.open sql,conn,1,3                            '执行查询，并将结果保存在 rs 中
        if not (rs.eof and rs.bof) then                 '存在满足条件的记录
                rs("comment")=request.form("comments")  '留言内容
                '回复的留言内容，replace 函数是用<BR> 替换回复内容中的 vbCRLF 字符
                rs("Reply")=replace(request.form("Reply"),vbCRLF,"<BR>")
                rs("ReplyDate") = Now()                  '回复时间
                rs("ReplyName") = Session("user")        '回复留言的管理员姓名
                rs.update                                '更新数据库
        end if
        rs.close                                         '关闭 RecordSet 对象
        '修改或回复成功，给出提示
        response.write "<script>alert('留言已经修改或回复成功，请单击“确定”返回！');</script>"
        response.redirect "admin_main.asp"              '返回到留言管理界面
        response.end
end if
'显示详细资料
set rs = server.createobject("adodb.recordset")
sql = "select * from note_comment where ID="&id
rs.open sql,conn,1,1
if rs.eof and rs.bof then
        response.write "<script>alert('无此留言，请单击“确定”返回！');</script>"
        response.redirect "admin_main.asp"
        response.end
end if
if not (rs.eof and rs.bof) then
        Comments=replace(rs("Comment"),"<BR>",vbCRLF)
        if rs("reply")<>"" then
        reply=replace(rs("Reply"),"<BR>",vbCRLF)
        else
        reply=""
        end if
        %>
```

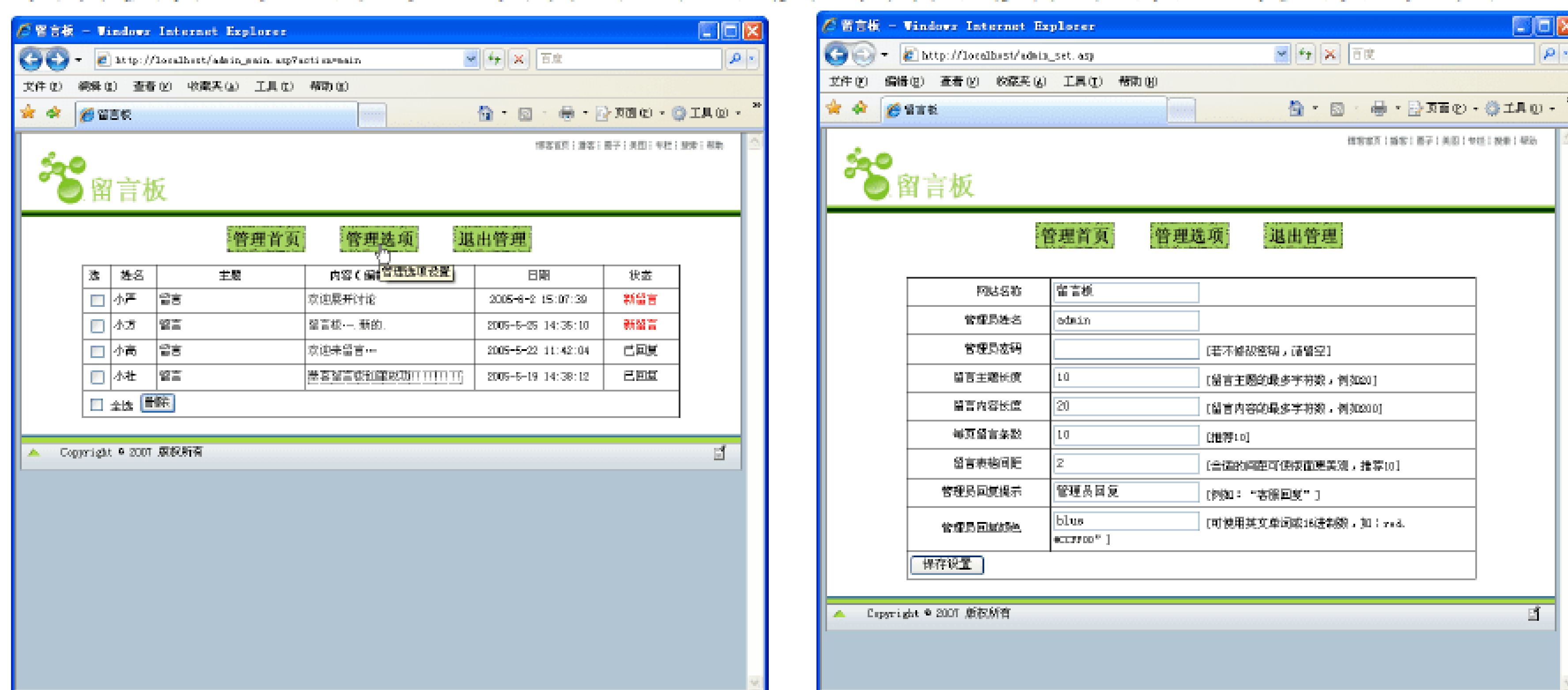
```
<tr><td align = center>
<table width="600" border="1" cellpadding="3" bordercolor="#333333" style="border-collapse:
    collapse;">
  <form name="reply" method="post" action='admin_main.asp?action=reply&id=<%=id%>'>
  <tr><TD align="right" width=20% height=15>留言者 IP 地址
      </TD><td><%=rs("IP")%></td></tr>
  <tr><TD align="right" width=20%>留言日期</TD><td><%=rs("PostDate")%></td></tr>
  <tr><TD align="right" width=20%>留言人姓名
      </TD><td><%=rs("Name")%> </td></tr>
  <tr><TD align="right" width=20%>留言邮箱
      </TD><td><%=rs("EMail")%> </td></tr>
  <tr><TD align="right" width=20%>留言网址</TD><td><%=rs("url")%> </td></tr>
  <tr><TD align="right" width=20%>其它联系方式
      </TD><td><%=rs("communication")%> </td></tr>
  <tr><TD align="right" width=20%>留言主题
      </TD><td><%=rs("subject")%> </td></tr>
  <tr><TD align="right" width=20%>内容</TD><td><textarea style="overflow:auto"
      name="comments" cols="60" rows="8"><%=Comments%></textarea></td></tr>
  <tr><TD align="right" width=20% valign=top>回复内容</TD><td><textarea
      style="overflow:auto" name="Reply" cols="60" rows="8"><%=reply%></textarea> </td></tr>
  <TR><TD align="right" width=20%> <INPUT TYPE="hidden" name=send
      value=ok></TD><TD>
  <input type="submit" name="action" value=" 提 交 "></TD></TR>
  </form></TABLE></td><tr>
<%
End If
rs.Close
Set rs = nothing
End Sub
%>
<table width="760" border=0 cellspacing=0 cellpadding=0 bgcolor="#FFFFFF" align=center
    class="grayline">
<tr><td>  </td></tr>
<tr><td width=30 height=37 background="IMAGES/down.gif"><a href=#top><img
    src=images/up.gif border=0></a></td>
<td height=37 background="IMAGES/down.gif">  Copyright &copy; 2007  版权所有
    </td>
<td width=30 height=37 background="IMAGES/down.gif"><a href=admin_login.asp title="管理留
    言本"><img src=images/admin.gif border=0></a></td>
</tr>
</table>
</body>
<%
```

```
conn.CLose
Set conn = nothing
%>
```

(2) 将 admin_main.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

7. 创建留言设置页面

留言设置页面 admin_set.asp 是留言板管理员设置留言板中每条留言的参数的页面。在该页面中，留言板管理员可以设置留言的主题长度、内容长度以及数量等参数，如图 13-42 所示。



单击“管理选项”按钮　　　　留言设置页面

图 13-42　打开留言设置页面

【练习 13-19】创建留言板的留言设置页面。

(1) 创建留言板设置页面文件 admin_set.asp，其代码如下所示：

```
<!--#include file="conn.asp"-->
<%
If Request.QueryString("action")="setup" then
 call setup()
Else
 If Session("id")<>"" Then
        Set rs = server.createobject("ADODB.RecordSet")
        rs.Open "Select * from note_Admin where id="&Session("id"),conn,1,3
%>
<HTML><HEAD>
<TITLE><%=sitename%></TITLE>
<link rel="stylesheet" href="book.css" type="text/css">
</HEAD>
<body background=images/bg.gif topmargin="0" leftmargin="0">
<table width="760" border=0 cellspacing=0 cellpadding=0 bgcolor="#FFFFFF" align=center
    class="grayline">
<tr><td><img border="0" src="IMAGES/TOPS.gif" width=758></td></tr>
</table>
```

```
<center>
<table width="760" border=0 cellspacing=0 cellpadding=0 align=center bgcolor="#FFFFFF"
      class="grayline">
<tr><td align=center height=50>
<a href=admin_main.asp><img border=0 src=images/admin_home.gif title="管理首页"></a>
         
<a href=admin_set.asp?action=setup><img border=0 src=images/admin_setup.gif title="管理选项设
      置"></a>    
<a href=admin_logout.asp?action=loginout><img border=0 src=images/loginout.gif title="管理结束
      后，一定要点击这里"></a>
</td></tr>
<tr><td align=center>
<br>
</td></tr>
<tr><td>
</center>
<center>
<table width="594" border="1" cellpadding="3" bordercolor="#333333" style="border-collapse:
     collapse">
 <form name=book action=admin_set.asp?action=setup method=post>
 <tr><td   width=141 align=right>网站名称      </td><td width="429"><input
     type=text name=sitename value="<%=sitename%>" maxlength=20></td></tr>
 <tr><td   width=141 align=right>管理员姓名      </td><td width="429"><input
     type=text name=admin value="<%=rs("admin")%>" maxlength=16></td></tr>
 <tr><td   width=141 align=right>管理员密码      </td><td width="429"><input
     type=password name=pwd value="" maxlength=16> [若不修改密码，请留空]</td></tr>
 <tr><td   width=141 align=right>留言主题长度      </td><td width="429"><input
     type=text name=subjectmaxlength value="<%=subjectmaxlength%>" maxlength=3> [留言主
     题的最多字符数，例如 20]</td></tr>
 <tr><td   width=141 align=right>留言内容长度      </td><td width="429"><input
     type=text name=maxlength value="<%=maxlength%>" maxlength=3> [留言内容的最多字符
     数，例如 200]</td></tr>
 <tr><td   width=141 align=right>每页留言条数      </td><td width="429"><input
     type=text name=counts value="<%=counts%>" maxlength=2>   [推荐 10]</td></tr>
 <tr><td   width=141 align=right>留言表格间距      </td><td width="429"><input
     type=text name=distance value="<%=distance%>" maxlength=2> [合适的间距可使版面更美
     观，推荐 10]</td></tr>
 <tr><td   width=141 align=right>管理员回复提示      </td><td width="429"><input
      type=text name=isreply value="<%=rs("isreply")%>" maxlength=10> [例如： “客服回
      复”]</td></tr>
 <tr><td   width=141 align=right>管理员回复颜色      </td><td width="429"><input
       type=text name=replycolor value="<%=rs("replycolor")%>" maxlength=10> [可使用英文单
       词或 16 进制数，如：red、#CCFF00”]</td></tr>
```

```
 <tr><td colspan=2 width="580"><INPUT name="save" TYPE="hidden" value="ok"><INPUT
           TYPE="submit" value="保存设置"></td></tr>
 </form>
</table>
</center>
</td></tr>
</table>
<table width="760" border=0 cellspacing=0 cellpadding=0 bgcolor="#FFFFFF" align=center
           class="grayline">
<tr><td>   </td></tr>
<tr><td width=30 height=37 background="IMAGES/down.gif"><a href=#top><img
           src=images/up.gif border=0></a></td>
<td height=37 background="IMAGES/down.gif">  Copyright &copy; 2007   版权所有
           </td>
<td width=30 height=37 background="IMAGES/down.gif"><a href=admin_login.asp title="管理留
           言本"><img src=images/admin.gif border=0></a></td>
</tr>
</table>
</body>
</html>
<%
        rs.close
        set rs=nothing
 Else
        response.write "<script>alert('当前没有登录用户');history.back();</script>"
        response.end
 End If
End if
Sub setup()
 if request("save")="ok" then                                    '保存设置
        Set NArs=Server.CreateObject("ADODB.Recordset")           '创建 RecordSet 对象
        '从数据库 Note_Admin 中查询 id 值为 Session("id")的记录，并保存在 NArs 中
        NArs.open "select * from note_admin where id="&Session("id"),conn,1,3

        Set WBrs = Server.CreateObject("ADODB.RecordSet")         '创建 RecordSet 对象
        '从数据库 Web 中查询所有记录，并保存在 WBrs 中
        WBrs.Open "Select * from web",conn,1,3

        '资料填写不完整
        if trim(request.form("sitename"))="" or trim(request.form("admin"))="" or
             trim(request.form("maxlength"))="" or trim(request.form("counts"))="" or
             trim(request.form("replycolor"))="" or trim(request.form("isreply"))="" or
             trim(request.form("distance"))="" then
```

```
            '给出提示，并返回到前一页
            response.write "<script>alert('资料填写不完整或有错误，请检查后重新提交。
              ');history.back();</script>"
            response.end
        end if
        '输入的留言主题最大长度、留言内容最大长度、每页留言数和表格间距不是数字
        if (not isNumeric(request.form("subjectmaxlength"))) or (not
                isNumeric(request.form("maxlength"))) or (not isNumeric(request.form("counts"))) or (not
                isNumeric(request.form("distance"))) then
            '给出提示，并返回到前一页
            response.write "<script>alert('资料填写不完整或有错误，请检查后重新提交。
                  ');history.back();</script>"
            response.end
        end if
        WBrs("sitename")=Request.Form("sitename")                         '站点名称
        NArs("admin")=Request.Form("admin")                               '管理员姓名
        if trim(request("password"))<>"" then
            NArs("pwd")=trim(Request.Form("pwd"))                         '密码
        End If
        WBrs("SubjectMaxLength") = Request.Form("subjectmaxlength")       '留言主题最大长度
        WBrs("maxlength")=Request.Form("maxlength")                       '留言内容最大长度
        WBrs("counts")=Request.Form("counts")                             '每页留言数
        WBrs("mail")=Request.Form("mail")                                 '是否必填邮箱
        NArs("isreply")=trim(Request.Form("isreply"))                     '回复提示
        NArs("replycolor")=trim(Request.Form("replycolor"))               '回复颜色
        WBrs("distance")=Request.Form("distance")                         '表格间距
        NArs.update                                                       '更新数据库
        WBrs.update                                                       '更新数据库
        NArs.Close                                                        '关闭 RecordSet 对象
        Set NArs = nothing
        WBrs.Close                                                        '关闭 RecordSet 对象
        Set WBrs = nothing
        '保存成功，给出提示，并返回到管理选项界面
        response.write "<script>alert('您提交的设置已保存。
          ');location.href='admin_set.asp';</script>"
        response.end
 End If
End Sub
conn.CLose
'关闭数据库连接
Set conn = nothing
%>
```

(2) 将 admin_set.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

8. 创建退出登录文件

用户可以参考下面练习所介绍的方法创建留言板管理员退出登录页面。

【练习 13-20】创建留言板管理员退出登录页面。

(1) 创建留言板管理员退出登录页面 admin_logout.asp，其代码如下所示：

```
<%
'退出管理，返回留言首页
Session("login") = ""                  '清除 Session 变量 login 中的值
Session("id") = ""                     '清除 Session 变量 id 中的值
Session("user") = ""                   '清除 Session 变量 user 中的值
Response.Redirect ("index.asp")        '页面转到留言查看界面
%>
```

(2) 将 admin_logout.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

13.2.4　实例测试

完成本节实例的操作后启动 IE 浏览器，在地址栏输入 http://localhost/index.asp 并按下 Enter 键显示留言板主界面，如图 13-43 所示。留言发布者可以在该页面中单击“发布留言”按钮，在打开的留言发布页面中发布新的留言，如图 13-44 所示。



图 13-43　留言板主界面



图 13-44　发布留言页面

单击留言板主界面右下角的“管理留言板”按钮，在打开的如图 13-45 所示的管理员登录页面中，登录留言板管理页面(用户名为 admin，密码为 admin)，如图 13-46 所示。

在如图 13-46 所示的留言板管理页面中，留言板管理员可以通过单击具体的留言名称(如图 13-47 所示)，打开如图 13-48 所示留言回复页面。在留言回复页面中，留言板管理员可以对留言进行回复。

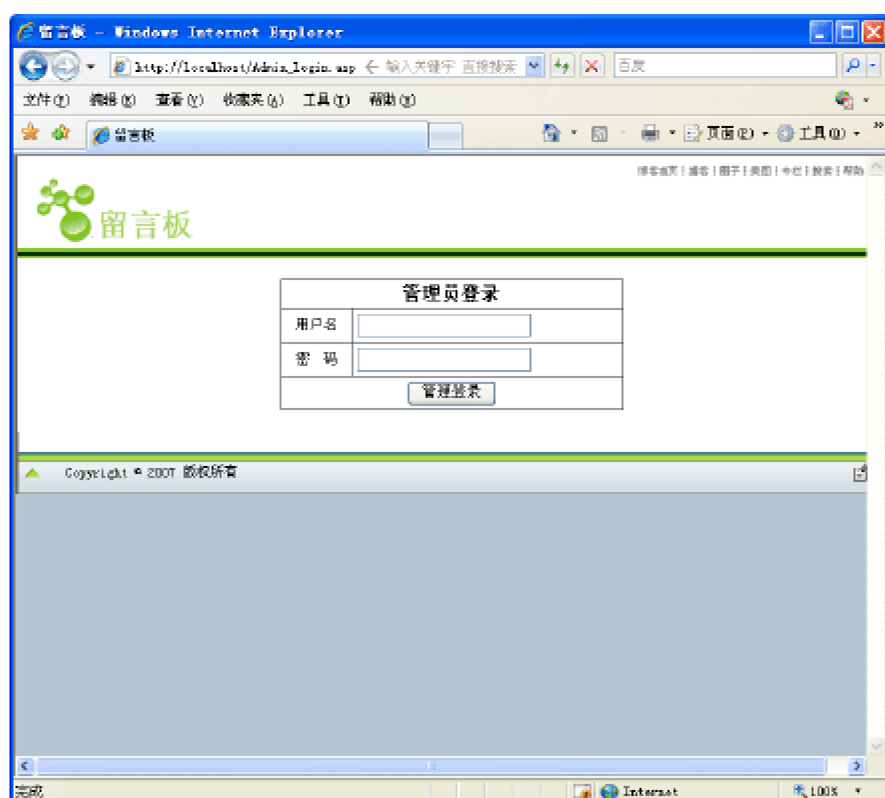


图 13-45　管理员登录页面



图 13-46　留言板管理页面



图 13-47　单击留言名称

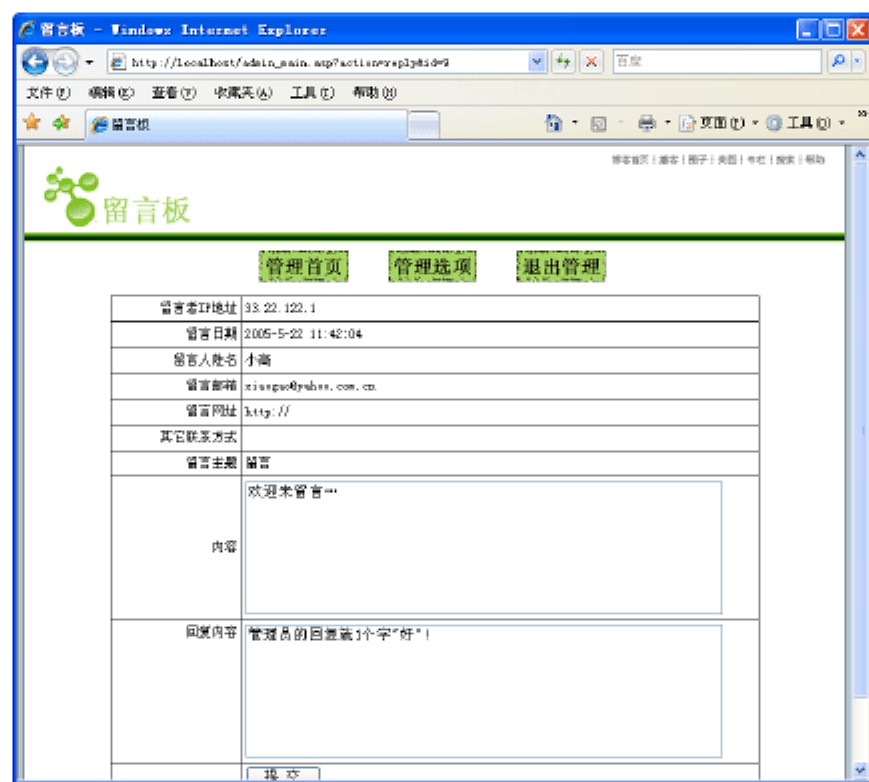


图 13-48　留言回复页面

在如图 13-46 所示的留言板管理页面中，留言板管理员还可以通过单击“管理选项”按钮(如图 13-49 所示)，打开如图 13-50 所示的留言板设置页面。在留言板设置页面中，留言板管理员可以根据需要设置留言板。



图 13-49　单击“管理选项”按钮



图 13-50　留言板设置页面

在如图 13-46 所示的留言板管理页面中，留言板管理员可以通过单击“退出管理”按钮，退出留言板管理界面。

13.3　网站流量统计

网站流量统计是现在很多网站和论坛必备的模块。该模块可以有效地统计网站的本日访问人数、历史访问人数以及平均访问人数等流量数据，可以帮助网站管理员对网站进行网站更新以提高网站访问量。

13.3.1　功能介绍

本节实例创建的网站流量统计模块如图 13-51 所示。单击该页面中的“网站流量详细数据查看”链接，打开如图 13-52 所示的网站详细流量查看页面，可以查看网站流量的详细数据。

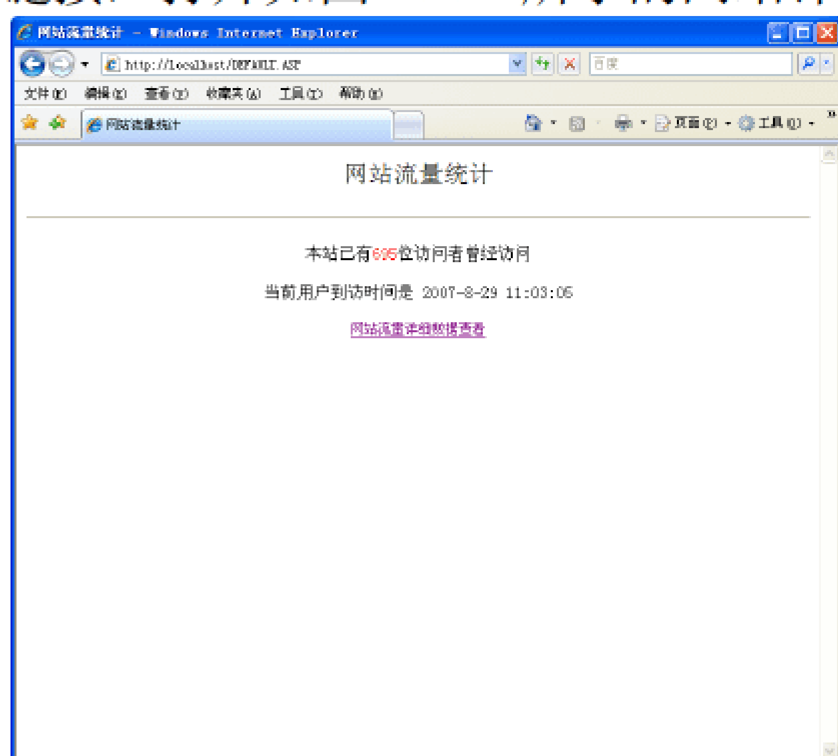


图 13-51　网站流量统计主页面

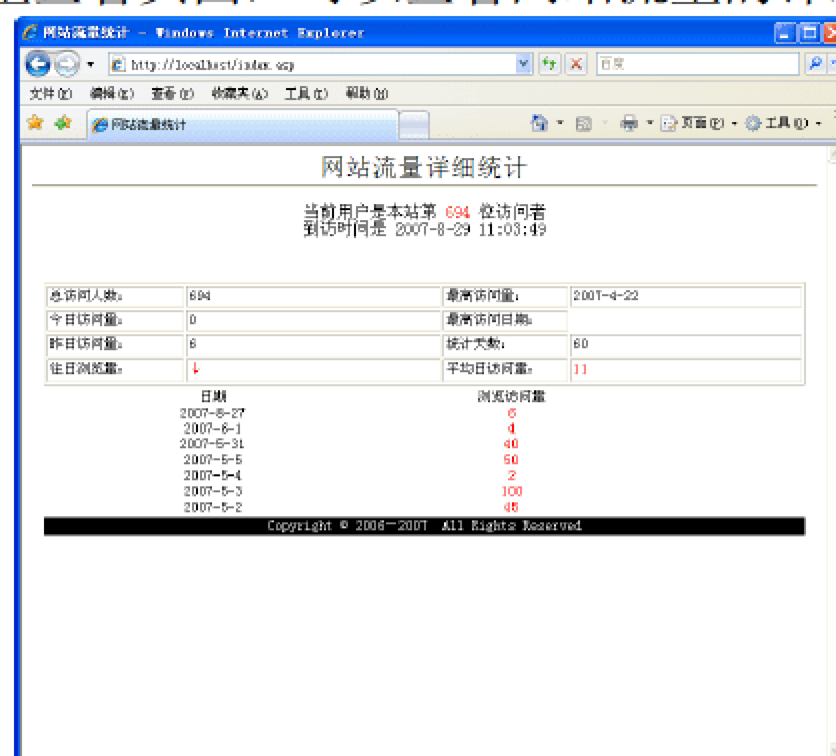


图 13-52　网站流量详细数据

13.3.2　模块设计

本节实例所创建的网站流量统计模块能够实现网站流量总访问量统计、网站每日平均访问量统计、当天网站访问量统计、网站最高访问量和最高访问日期统计等功能，其工作原理如图 13-53 所示。

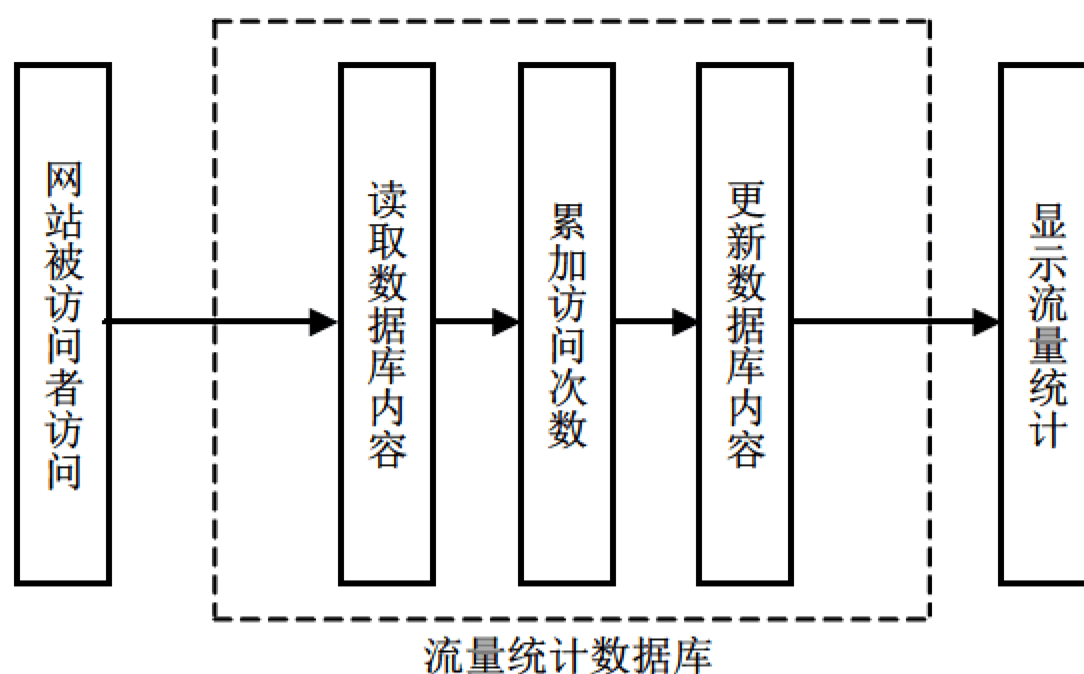


图 13-53　网站流量统计模块的流程

1. 页面文件设计

由网站流量计数器模块的结构可以看出，该模块主要由以下几个页面文件组成。

- Count.asp 文件：该文件为数据库连接文件。
- Default.asp 文件：该文件向登录用户显示当前访问总量以及用户的登录时间。
- Index.asp 页面：该文件详细显示本日网站流量统计和 10 条最近的历史流量统计。

2. 数据库设计

本节实例所创建的网站流量统计模块需要如表 13-5 所示的数据表 count 存储网站各项访问量的数据，需要如表 13-6 所示的数据表 history 存储日期和访问量数据。

表 13-5　count 数据表

字　　段	字 段 类 型	主 关 键 字	说　　明
ID	自动编号	*	编号
Today	整型	—	今日访问量
Ota	日期时间	—	日期
daynum	整型	—	统计天数
yesterday	整型	—	昨日访问量
Counter	整型	—	总访问量
Top	整型	—	最高访问量
Topdate	日期时间	—	最高访问日期

表 13-6　history 数据表

字　　段	字 段 类 型	主 关 键 字	说　　明
ID	自动编号	*	编号
Date	日期时间	—	日期
Browser	整型	—	访问量

13.3.3　实例操作

下面将结合本章第 13.2.2 节所介绍的网站流量统计模块的结构，通过实例介绍创建一个网站流量统计模块方法。用户在进行实例操作之前，应结合本书第 1.2 节和第 1.3 节所介绍的方法在本地计算机上构建一个 ASP 应用程序的开发环境。

1. 创建数据库

下面将根据表 13-5 和表 13-6 所介绍的内容，创建本节实例所需的数据库。

【练习 13-21】根据表 13-5 和表 13-6 所示的内容创建数据表 count 和 history。

(1) 选择“开始”|“程序”|Microsoft Office| Access 2003 命令，启动 Access 2003 数据

库，然后选择“文件”|“新建”命令，打开“新建文件”对话框。

(2) 单击“空数据库”按钮，在打开的“文件新建数据库”对话框的“文件名”文本框中输入 counter.mdb。

(3) 单击“创建”按钮，在打开的 Access 2003 的工作界面的工作区域中选择“对象”列表框中的“表”选项，然后在对话框右侧的选项区域中双击“使用设计器创建表”选项打开数据表的设计检视窗口。

(4) 根据表 13-5 所示的内容，创建数据表 count，完成后的效果如图 13-54 所示。

(5) 根据表 13-6 所示的内容，创建数据表 history，完成后的效果如图 13-55 所示。

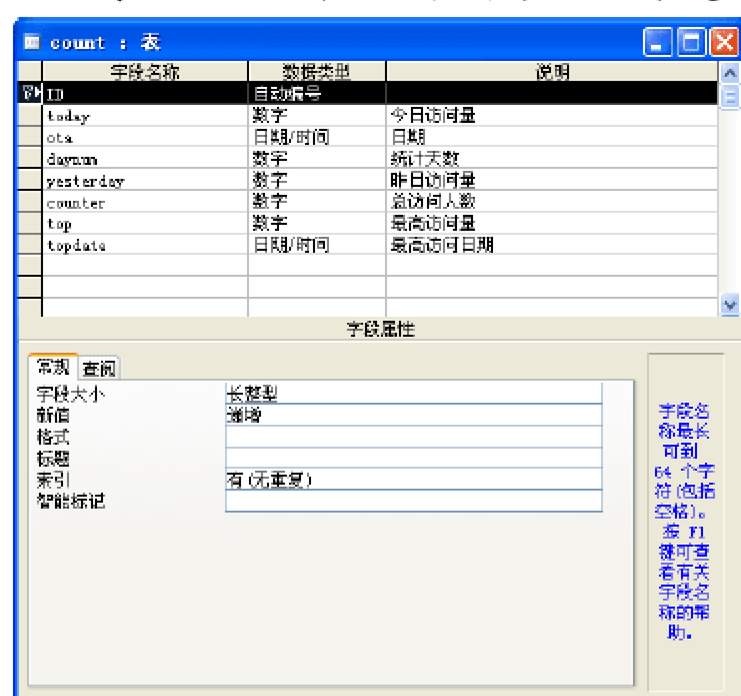


图 13-54　创建 count 数据表

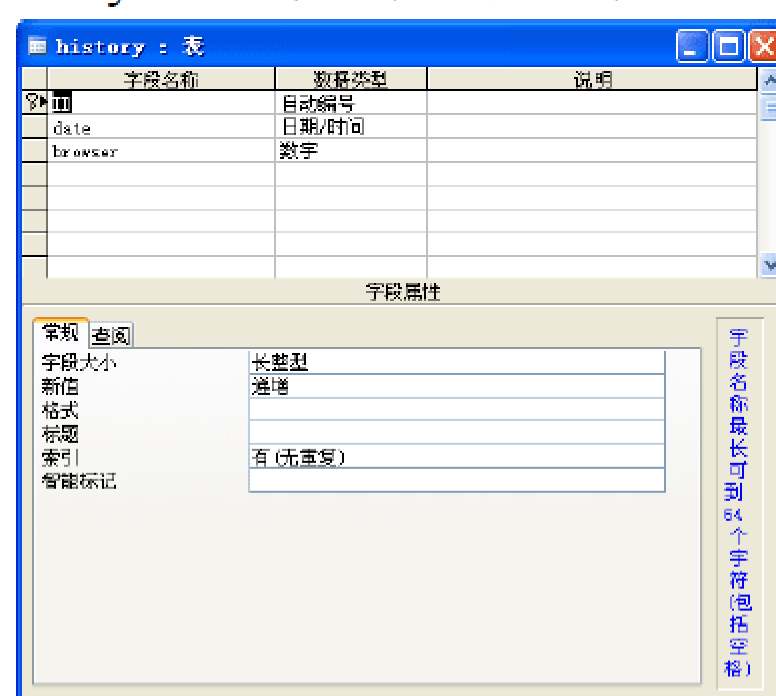


图 13-55　创建 history 数据表

(6) 完成以上操作后关闭 Access 数据库。

2. 创建数据库连接文件

完成数据库的创建工作后，用户可以参考下面练习所介绍的方法，创建 ASP 网页数据库连接文件 Conn.asp。

【练习 13-22】面。

(1) 创建数据库连接文件 Conn.aso，其代码如下所示：

```
<%
'使用 createobject 方法创建 connection 对象
set conn=server.createobject("adodb.connection")
'Server.MapPath 将指定的数据库文件虚拟路径转换为真实路径，本例对应的路径为
'd:\Inetpub\wwwroot\count\counter.mdb
DBPath = Server.MapPath("counter.mdb")
'采用 Open 方法连接数据库，driver 参数设置数据库的驱动程序，
' dbq 参数设置数据库的实际路径及文件名称
conn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" & DBPath
set rs=server.createobject("adodb.recordset")            '创建 recordset 数据集对象
sql="select * from count"                                '从 count 表中读取所有数据
rs.open sql,conn,1,3                                     '执行数据库查询操作
Dim vtoday,vbrowser,vota,vdaynum,vyesterday,vcounter,vtop,vtopdate
```

```
'rs("today")、rs("ota")、rs("daynum")、rs("yesterday")、rs("counter")、rs("top")、'rs("topdate")分别为
count 表中 today 列、ota 列、daynum 列、yesterday 列、counter 列、top 列、'topdate 列对应的值
vtoday = rs("today")
vota = rs("ota")
vdaynum = rs("daynum")
vyesterday = rs("yesterday")
vcounter = rs("counter")
vtop = rs("top")
vtop = rs("topdate")
%>
```

(2) 将 Conn.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

3. 创建计数器主页面

用户访问网站首先要进入一个默认网页 Default.asp，该页面显示访问计数器以及用户登录该网站的时间，并具备数据库的更新操作。

【练习 13-23】创建网站计数器主页面 Default.asp。

(1) 创建网站流量计数器主页面文件 Default.asp，其代码如下所示：

```
<!--#include file="conn.asp"-->
<html>
 <head>
  <title>网站流量统计</title>
 </head>
<body bgcolor="red">
<p align="center"><font size="5">网站流量统计</font></p>
<hr>
<p align="center">本站已有<font color= red><%=vcounter+1%></font>位访问者曾经访问</p>
<%                                          '对数据库的更新操作
If vota<>date() Then                        'count 表中的数据不是本日数据
          vzzday= vota                      '获得 count 表中的日期
     '将 count 表更新为本日数据，今日访问量置为 0
     conn.Execute"Update count set today=0, ota=date(),daynum=daynum+1,
         yesterday="&vtoday&""
    在'history 数据表中添加记录
    '创建 RecordSet 对象
    Set rs = Server.CreateObject("ADODB.RecordSet")
    '从数据库中读取所有记录
    rs.Open "Select * from history",conn,1,3
    rs.AddNew
    '使用 AddNew 方法添加一条新记录
    rs("date") = vzzday                     '日期
    rs("browser") = vtoday                  '浏览量
    rs.Update                               '更新数据库
    rs.Close                                '关闭 RecordSet 对象
```

```
Else                                              'count 表中保存的是今日数据
       If vtop<vtoday Then                        '最高访问量低于今日访问量
           '更新 count 表中的值，将今日访问量、总访问量加 1，最高访问量更新为今日访问量
           conn.Execute"Update count set counter=counter+1,today=today+1,top = "&vtoday&""
       Else                                       '最高访问量高于今日访问量
      '更新 count 表中的值，将今日访问量、总访问量加 1
      conn.Execute"Update count set counter=counter+1,today=today+1"
       End If
End If
%>
<p align="center">当前用户到访时间是 <%=Date()%> <%=Time()%></p>
<p align="center"><font size="2"><a href = "index.asp">网站流量详细数据查看
     </a></font></p>
</body>
</html>
```

(2) 将 Index.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

13.3.4　实例测试

完成本节实例的操作后启动 IE 浏览器，然后在地址栏中输入 http://localhost/default.asp 并按下 Enter 键，网页效果如图 13-56 所示。在 default.asp 页面中单击“网站详细数据查看”链接后，可以打开如图 13-57 所示的 Index.asp 页面显示网站的详细访问数据。



图 13-56　default.asp 页面效果

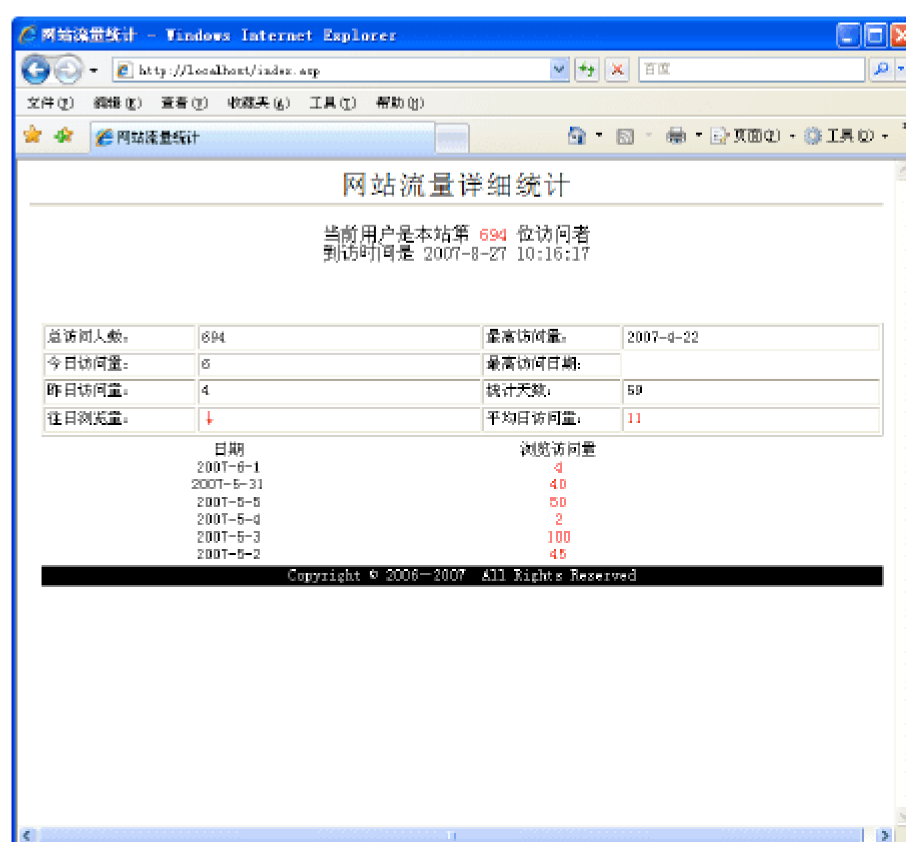


图 13-57　Index.asp 页面效果

13.4　网络聊天室

本节将通过一些练习，介绍使用 ASP 编程的方式创建一个网络聊天室的方法。在聊天室中，用户能够实现最基本的聊天功能。

13.4.1　功能介绍

本节创建的网络聊天室的登录界面如图 13-58 所示，用户可以在该界面中登录聊天室或注册新的聊天室用户。成功登录聊天室后，用户将进入如图 13-59 所示的聊天室主界面。用户可以在聊天室主页面下方的文本框中输入和发送聊天内容，可以在页面左侧显示聊天室中的全部用户，并在页面的中间部分查看聊天室中的聊天内容。



图 13-58　聊天室登录页面

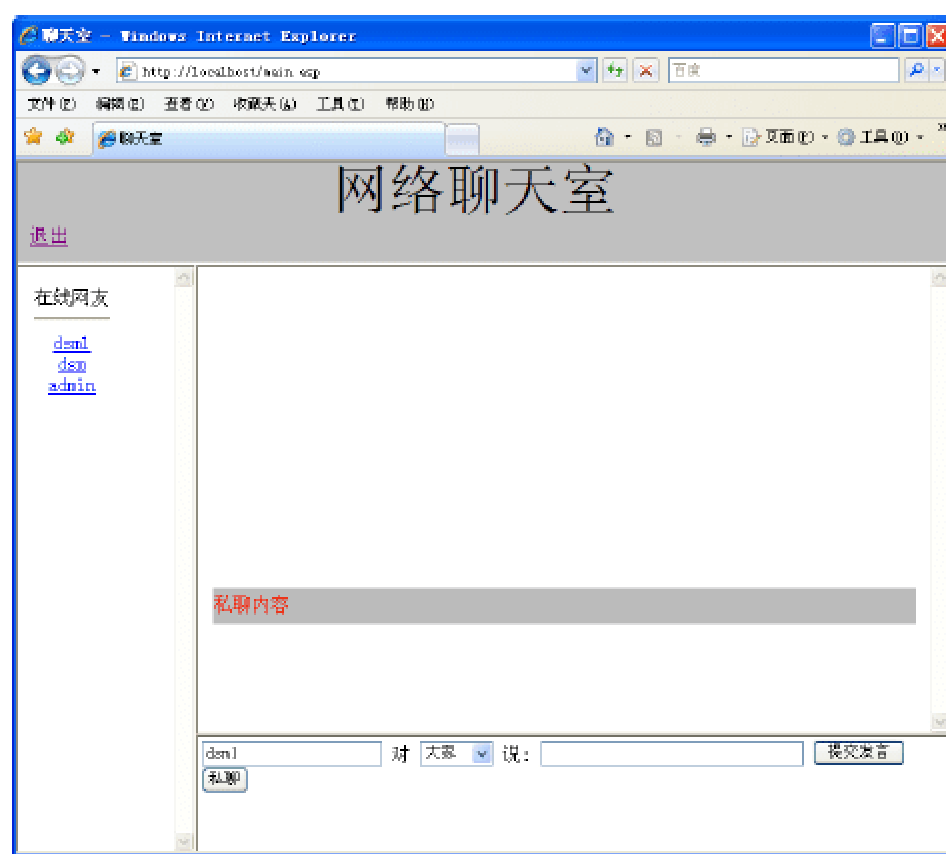


图 13-59　聊天室主界面

13.4.2　模块设计

本节实例创建的网络聊天室模块的结构如图 13-60 所示。

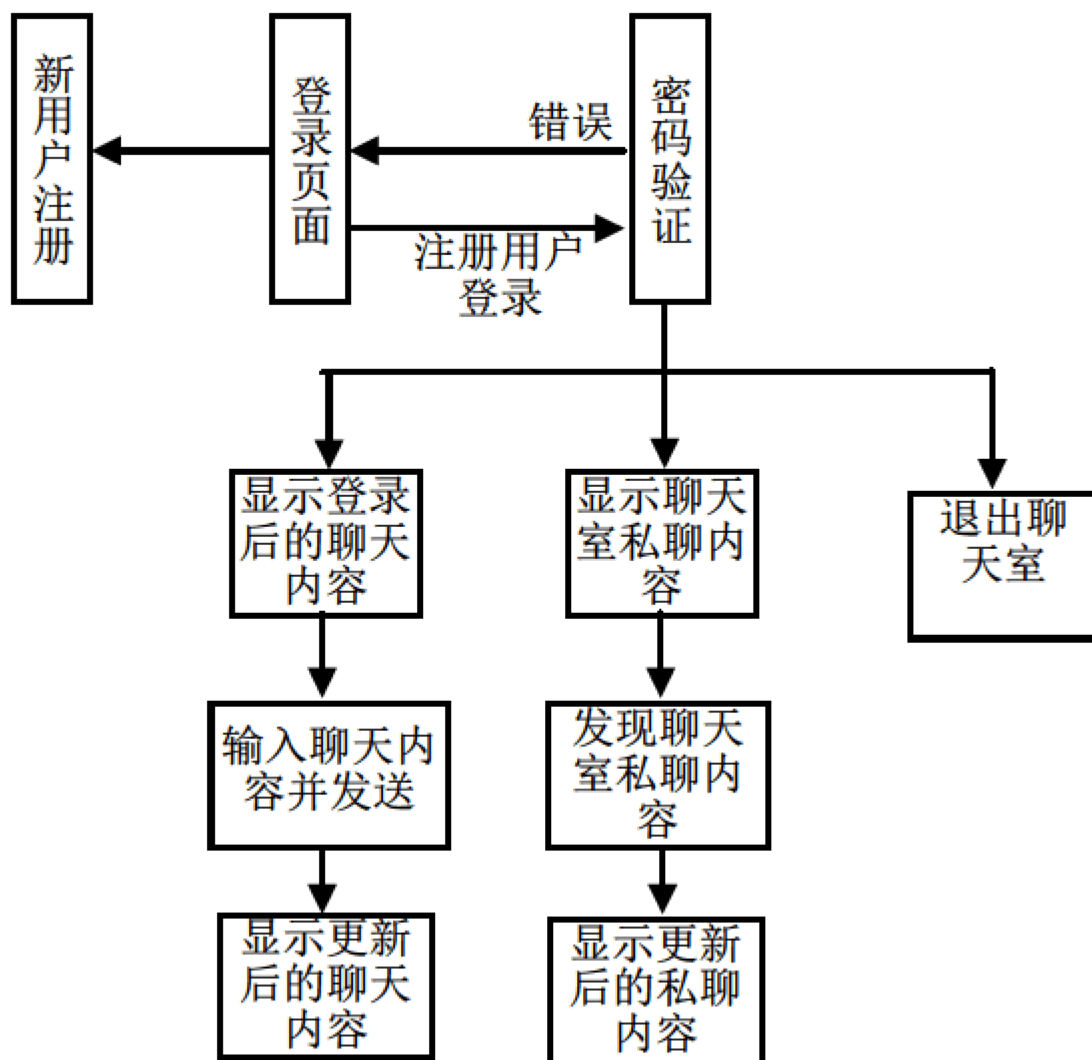


图 13-60　网络聊天室模块的结构

1. 页面文件设计

由网站聊天室模块的结构可以看出，该模块主要由以下几个页面文件组成。

- Index.asp 页面：该页面为聊天室用户登录页面。
- Register.asp 页面：该页面为聊天室新用户注册页面。
- Main 页面：该页面为聊天室系统的主界面。
- Chatroom.asp 文件：该文件为聊天室系统的布局文件。
- Showname.asp 页面：该页面显示当前在线用户名单。
- Postmsg.asp 文件：该文件实现用户输入和发送聊天内容。
- Showmasg.asp 文件：该文件显示用户登录后的所有聊天内容。
- Topmsg.asp 文件：该文件在聊天室主界面上方显示聊天室名称。
- Logout.asp 文件：该文件为用户退出登录文件。
- Conn.asp 文件：该文件为数据库连接文件。
- Md5.asp 文件：该文件为用户登录密码加密文件。
- Procedure.asp 文件：该文件中包含邮件地址格式确认过程。

另外，在本节创建的聊天室模块中需要处理以上所示的文件以外，还需要创建 Function.asp、Safecode.asp、Canvsa.asp 以及 font.asp 等 4 个文件完成用户登录时验证码的生成与显示。

2. 数据库设计

本章所创建的网络聊天室模块需要使用如表 13-7 所示的 User 数据表保存用户注册信息，需要如表 13-8 所示的 online 数据表保存用户注册信息，需要如表 13-9 所示的 chat 数据表保存用户的聊天内容。

表 13-7　User 数据表的结构

字　段	字段类型	主 关 键 字	说　明
ID	自 动 编 号	*	编号
UserName	文本	—	用姓名户名
Password	文本	—	用户密码
Name	文本	—	用户真实姓名
Email	文本	—	用户电子邮箱

表 13-8　online 数据表的结构

字　段	字 段 类 型	主 关 键 字	说　明
ID	自动编号	*	编号
UserName	文本	—	用户名
OnlineTime	日期时间	—	登录时间

表 13-9　chat 数据表

字　段	字段类型	主关键字	说　明
ID	自动编号	*	编号
FromName	文本	—	发言者用户名
ToName	文本	—	发言对象用户名
PostTime	日期时间	—	发言时间
Chat	文本	—	发言内容

13.4.3　实例操作

下面将结合本章第 13.2.2 节所介绍的聊天室模块的结构，通过实例介绍创建一个网站聊天室模块的方法。用户在进行实例操作之前，应结合本书第 1.2 节和第 1.3 节所介绍的方法在本地计算机上构建一个 ASP 应用程序的开发环境。

1. 创建数据库

下面将根据表 13-7～表 13-9 所示的内容，创建本节实例所需的数据库。

【练习 13-24】根据表 13-7～表 13-9 所示的内容，创建数据表 User、online 和 chat。

(1) 选择“开始”|“程序”| Microsoft Office| Access 2003 命令，启动 Access 2003 数据库，然后选择“文件”|“新建”命令，打开“新建文件”对话框。

(2) 单击“空数据库”按钮，在打开的“文件新建数据库”对话框的“文件名”文本框中输入 ChatRoom.mdb。

(3) 单击“创建”按钮，在打开的 Access 2003 的工作界面的工作区域中选择“对象”列表框中的“表”选项，然后在对话框右侧的选项区域中双击“使用设计器创建表”选项，打开数据表的设计检视窗口。

(4) 根据表 13-7 所示的内容，创建数据表 User，完成后的效果如图 13-61 所示。

(5) 根据表 13-8 所示的内容，创建数据表 online，完成后的效果如图 13-62 所示。

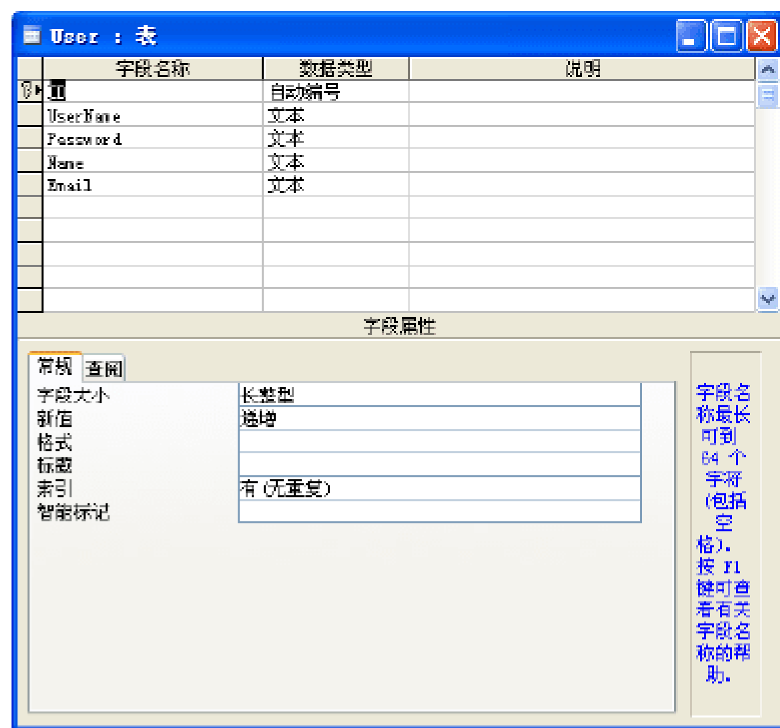


图 13-61　数据表 User

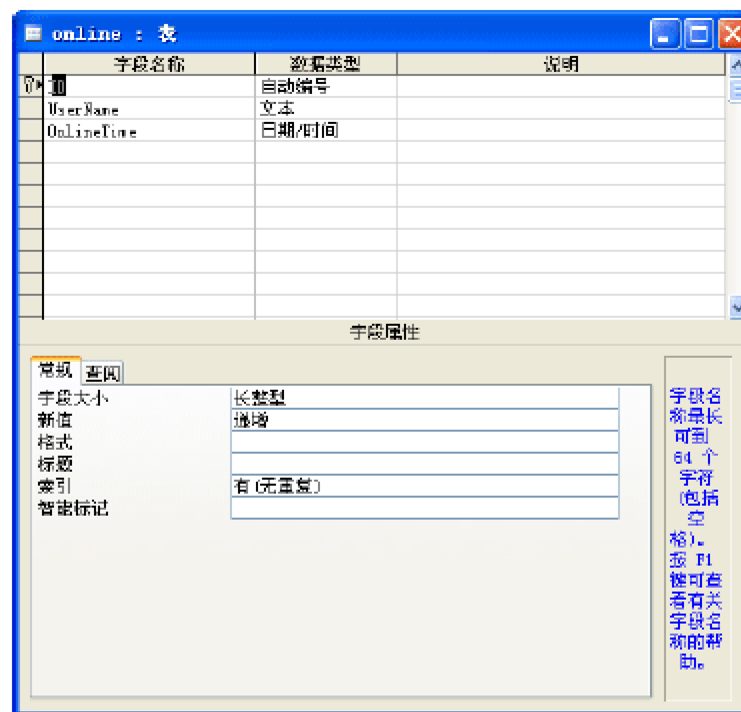


图 13-62　数据表 online

(6) 根据表 13-19 所示的内容，创建数据表 chat，完成后的效果如图 13-63 所示。

(7) 完成以上操作后 ChatRoom 数据库如图 13-64 所示。这时，用户可以关闭 Access 2003 数据库。

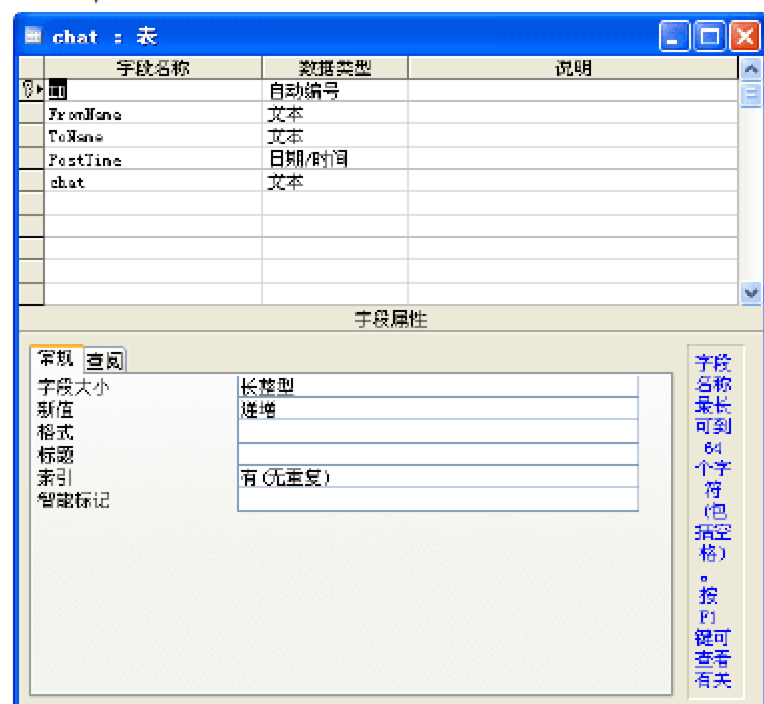


图 13-63　数据表 chat

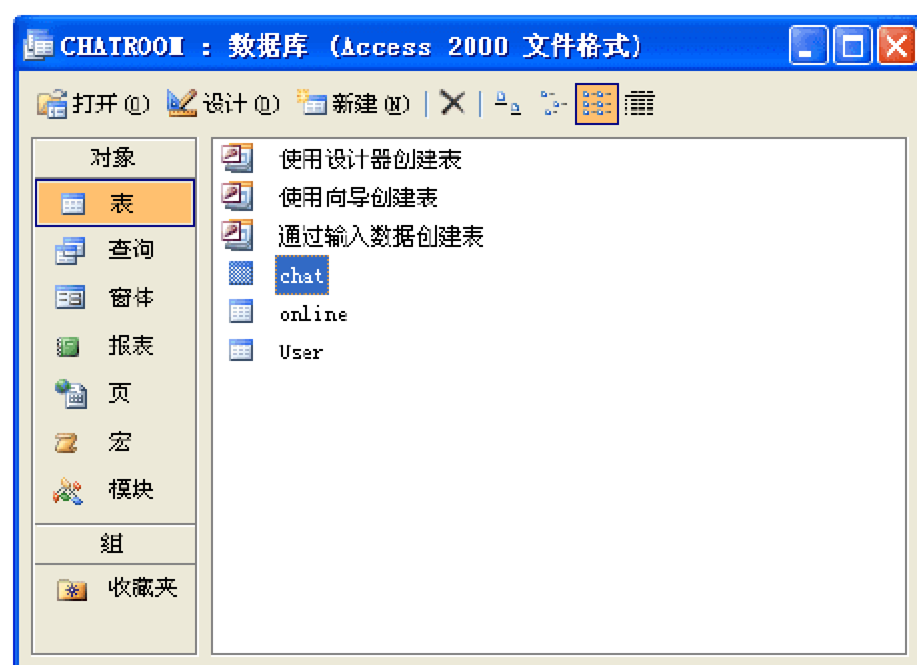


图 13-64　ChatRoom 数据库

2. 创建数据库连接文件

完成数据库的创建工作后，用户可以参考下面练习所介绍的方法，创建 ASP 网页数据库连接文件 Conn.asp。

【练习 13-25】创建数据库连接文件 Conn.asp。

(1) 创建数据库连接文件 Conn.asp，其代码如下所示：

```
<%
'连接数据库开始
'dim conn,connstr,database
'database = "chatroom.mdb"
'set conn=server.createobject("adodb.connection")
'connstr="DBQ="+Server.mappath(database)+";DRIVER={Microsoft Access Driver (*.mdb)};"
'conn.Open connstr
%>
<%
DBName="chatroom.mdb"
Set conn = Server.CreateObject("ADODB.Connection")
connstr="DBQ="+Server.mappath(DBName)+";DRIVER={Microsoft Access Driver (*.mdb)};"
conn.open connstr
%>
```

(2) 将 Conn.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

3. 创建聊天室登录模块

聊天室登录模块包括聊天室登录页面 Index.asp 和 Function.asp、Safecode.asp、Canvas.asp、font.asp 以及 MP5.asp 等用户登录验证码生成与显示文件。用户在创建本节实

例所需的聊天室登录模块时，可以先参考本章第 13.1.3 节相应的方法创建 Function.asp、Safecode.asp、Canvas.asp、font.asp、MP5.asp 文件，再参考下面练习所介绍的方法创建 Index.asp 页面。

【练习 13-26】创建聊天室登录主界面 Index.asp。

(1) 创建网络聊天室用户登录页面 Index.asp，其代码如下所示：

```
<!--#Include File="conn.asp" -->
<!--#Include File="md5.asp" -->
<!--#Include File="function.asp" -->
<%
If Request.QueryString("act") ="login" Then          '用户登录
 user = Request.Form("UserName")                     '读取从表单传递过来的用户名数据
pass = Request.Form("Password")                      '读取从表单传递过来的密码数据
   If user<>"" and pass<>"" Then                     '用户名和密码已经填写
If not IsNumeric(Request.Form("confirm")) Then
'用函数 IsNumeric 判断用户输入的验证码是否为数字
'如果用户填写的验证码不是数字，则提示用户并返回到登录窗口
Response.Write "<script>alert('你输入的验证码为非数字！');
    window.location.href('index.asp');</script>"
          Response.End
        End If
     'Session("SafeCode")用来保存程序生成的验证码，
     '比较生成的验证码与用户输入的验证码是否相同
     If (int(Session("SafeCode"))=int(Request.Form("confirm"))) Then
            Set rs=Server.CreateObject("Adodb.RecordSet")   '生成 RecordSet 对象
     '从数据表 User 中读取 UserName 的值为 user 和 UserPass 的值为 md5(pass)的记录
     'md5 函数是一个加密函数，将管理员输入的密码加密后与数据库中的值进行比较
          sql="Select * from User where UserName='"&user&"' and Password='"&md5(pass)&"'"
             rs.open sql,conn,1,3                       '读取满足条件的记录保存在 rs 中
          If not(rs.bof and rs.eof) Then                '满足条件的记录存在
                                                        '输入的用户名和密码正确
               If md5(pass)=rs("Password") and user=rs("UserName") Then
                 '判断登录的用户是否已在线
                 sql = "Select * from online where UserName = '"&user&"'"
            Set OLrs = Server.CreateObject("Adodb.RecordSet")
            OLrs.open sql,conn,1,1
            '用户已在线
            If not (OLrs.EOF and OLrs.BOF) Then
                 '给出提示，并返回到前一页
                 Response.Write("<script>alert('该用户已经存在！');history.back();</script>")
                 Response.End
            Else                                        '用户不在线
```

```
                    Session("user")=user  'Session 变量 user 中保存当前登录的用户名
                    Session("chat") = ""   'Session 变量 chat 保存用户私聊内容
                    'Application("Gchatnum")中的值为当前聊天总记录数
                    Session("Lchatnum") = Application("Gchatnum")
                    OLrs.close                     '关闭 RecordSet 对象
                    Set OLrs=nothing
                    rs.Close
                    Set rs = nothing
                    Response.Redirect "main.asp"       '登录成功，转到页面 main.asp
                    End If
                    Else
                    '登录不成功，提示用户，并转到页面 index.asp
                    Response.Write "<script>alert('登录失败，用户名或密码错误！
                        ');window.location.href('index.asp');</script>"
                    End If
                    conn.Close                      '关闭数据库连接
                    Set conn=nothing
            Else                                  '满足条件的记录不存在
                    Response.Write "<script>alert( '登录失败！用户名或密码不存在！
                        ');window.location.href('index.asp');</script>"
            End If
            conn.Close                            '关闭数据库连接
            Set conn=nothing
          Else                                    '验证码输入错误
            Response.Write "<script>alert('登录失败！验证码输入错误！');
               window.location.href('index.asp');</script>"
            End If
    Else                                          '用户名和密码没有输入
        Response.Write "<script>alert('登录失败！用户名和密码不能为空！
           ');window.location.href('index.asp');</script>"
    End If
Else
%>
<html>
<head>
<body bgcolor="#C0C0C0">
<title>网站聊天室</title>
<script language="VBScript">
function check()                          '检查是否输入用户名、密码和验证码
 If form1.username.value="" Then          '“用户名”文本框的值为空
          alert("用户名不能为空!")          '给出提示
          form1.username.focus()          '将“用户名”文本框置为输入焦点
          return false
```

```
 End If
 If form1.password.value="" Then              '“密码”文本框的值为空
               alert("密码不能为空!")
               form1.password.focus()          '将“密码”文本框置为输入的焦点
               return false
 End If
 If form1.confirm.value="" Then               '验“证码”文本框的值为空
               alert("验证码不能为空！")
               form1.confirm.focus()           '将“验证码”文本框置为输入焦点
               return false
 End If
 return true
End Function
 </script>
</head>
<body>
<p align="center"><font size="8" face="宋体">聊天室用户登录</font></p>
<hr>
<form name = "form1" method = "Post" action = "index.asp?act=login" onsubmit = "Check()">
<div align="center">
   <table border="0" width="37%">
      <tr>
         <td width="28%" bgcolor="#C0C0C0" align = center><font size="2">用户名：</font></td>
         <td width="72%"><input type = "input" name = "UserName" size = 20></td>
      </tr>
      <tr>
         <td width="28%" bgcolor="#C0C0C0" align = center><font size="2">密  码：
               </font></td>
         <td width="72%"><input type = "password" name = "Password" size = 20></td>
          </tr>
          <tr>
         <td width="28%" bgcolor="#C0C0C0" align = center><font size="2">验证码：</font></td>
         <td width="72%"><input type = "input" name = "Confirm" size = 10><%Call
             GetSafeCode()%></td>
</tr>
</table>
</div>
<b>
 <p align="center"><input type = "submit" name = "submit" value = "登录"> 
<input type = "reset" name = "submit" onclick = "check()" value = "重置"></p>
<p align="center"><a href="Register.asp">注册新用户</a></p>
</form>
```

```
</body>
<%End If%>
</html>
```

(2) 将 Index.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

4. 创建聊天室新用户注册模块

聊天室新用户注册模块包括电子邮件地址格式确认文件 Procedure.asp 和聊天室新用户注册文件 Register.asp。用户可以参考下面练习所介绍的方法创建聊天室新用户注册模块所需的页面。

【练习 13-27】创建电子邮件地址确认文件 Procedure.asp。

(1) 创建电子邮件地址确认文件 Procedure.asp，其代码如下所示:

```
<%
Dim webname,weburl,filepath,Language,errortext
'检查 Email 格式是否正确
Function IsValidEmail(email)                    '定义过程
Dim names, name, i, c
IsValidEmail = TRUE                             '初始化
'使用@字符将 Email 字符串分成几个子字符串并保存在 names 数组中
names = Split(email, "@")
'Ubound 函数返回数组 names 的最大下标，Ubound(names)<>1 表明 Email 字符串中存在@字符
'并不是一个，所以 Email 不是有效的邮件地址格式
If UBound(names) <> 1 Then
IsValidEmail = FALSE
    Exit Function                               '跳出 Function 过程
End If
For Each name in names                          '数组 names 中的每一个元素
    If Len(name) <= 0 Then                      '字符串 name 中字符的数目
    IsValidEmail = FALSE
    Exit Function                               '跳出 Function 过程
    End If
    For i = 1 To Len(name)                      'For 循环
    'Mid(name,i,1)返回字符串 name 中第 i 个字符，LCase 函数将该字符转换成小写形式
    c = LCase(Mid(name, i, 1))
    'InStr 函数返回某字符串在另一字符串中第一次出现的位置。
    'InStr("abcdefghijklmnopqrstuvwxyz_-.", c) <= 0 表明字符 c 不在字符串
    ' "abcdefghijklmnopqrstuvwxyz_-."中。IsNumeric(c)判断字符 c 是否为数字
    If InStr("abcdefghijklmnopqrstuvwxyz_-.", c) <= 0 AND NOT IsNumeric(c) Then
     '不支持中文格式地址
            IsValidEmail = FALSE
            Exit Function                       '跳出 Function 过程
```

```
            End If
            Next                                   '结束 For to 循环
            'left(name,1)返回字符串 name 最左边一个字符
            'Right(name,1) 返回字符串 name 最右边一个字符
            If Left(name, 1) = "." or Right(name, 1) = "." Then
                IsValidEmail = FALSE
                Exit Function                      '跳出 Function 过程
            End If
        Next                                           '结束 For each 循环
        If InStr(names(1), ".") <= 0 Then        'Email 字符串中@右边部分不包含字符“.”
            IsValidEmail = FALSE
            Exit Function
        End If
        'InStrRev 函数返回某字符串在另一个字符串中出现的从结尾计起的的位置，
        'InStrRev(names(1), ".")得到字符“.”在字符串 names(1)中从结尾计起的位置
        i = Len(names(1)) - InStrRev(names(1), ".")
        '电子邮件最后一般为 cn 或 com，长度为 2 或 3
        If i <> 2 AND i <> 3 Then
            IsValidEmail = FALSE
            Exit Function
        End If
        'Email 中存在字符串“..”
        If InStr(email, "..") > 0 Then
            IsValidEmail = FALSE
        End If
        End Function
        %>
```

(2) 将 Procedure.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

完成 Procedure.asp 文件的创建后，用户可以参考下面练习所介绍的方法创建聊天室新用户注册页面 Register.asp。

【练习 13-28】创建聊天室新用户注册页面 Register.asp。

(1) 创建网络聊天室的新用户注册页面 Register.asp，其代码如下所示。

```
<!--#Include File="conn.asp" -->
<!--#include File ="procedure.asp"-->
<!--#Include File="md5.asp" -->
<%
Sub save()
 '定义过程
```

```
Dim UserName,UserPwd,Pwd,Name,Email
'声明变量
'读取从表单传递过来的用户名数据，Trim 函数去掉字符串的前导与后续空格
 UserName=TRIM(Request.Form("UserName"))
 UserPwd=TRIM(Request.Form("UserPwd"))                 '读取从表单传递过来的密码数据
 Pwd=TRIM(Request.Form("Pwd"))                          '读取从表单传来的确认密码数据
 '读取从表单传递过来的真实姓名数据
 Name=TRIM(Request.Form("Name"))
 '读取从表单传递过来的电子邮箱数据
 Email=TRIM(Request.Form("Email"))
 '用户输入不完全
 If UserName="" or UserPwd="" or Pwd = "" or Name="" or Email="" Then
       '给出提示，返回注册页面。history.back()返回注册页面
       Response.Write "<script>alert('错误：请输入所有信息');history.back();</script>"
       Response.End
 End If
 if UserPwd<>Pwd Then                              '两次输入的密码不同
       '给出提示，并返回注册页面
       Response.Write "<script>alert('两次输入的密码不相同！');history.back(); </script>"
       Response.End
 End If
 'IsValidEmail 过程判断用户输入的发件人的邮件地址格式是否正确
 If NOT IsValidEmail(Email) Then
       '邮件地址格式错误，则给出提示，并返回到注册页面
       Response.Write "<script>alert('错误：邮件地址格式错误!');history.back();</script>"
       Response.End
 End If
 '判断数据库中是否存在用户输入的用户名
 Set HVrs = Server.Createobject("adodb.recordset")
 '从数据库中查找 UserName 值为 UserName 的记录
 sql = "Select * from user where UserName = '"&UserName&"'"
 HVrs.Open sql,conn,1,1
 If not (HVrs.EOF and HVrs.BOF) Then          '存在满足条件的记录
       '用户名在数据库中存在，则给出提示，并返回注册页面
       Response.Write "<script>alert('该用户名已经存在！请重新填写用户名
          ');history.back();</script>"
       Response.End
 Else                                          '不存在满足条件的记录
       '在数据库中插入一条新记录
       Set rs=Server.Createobject("adodb.recordset")
       sql="select UserName,Password,Name,Email from User"
       rs.open sql,conn,1,3
       rs.AddNew
```

```
            rs("UserName")=UserName             '用户登录名
            rs("Password")=md5(UserPwd)         '用户登录密码
            rs("Name")=Name                     '用户真实姓名
            rs("Email")=Email                   '用户电子邮箱
            rs.update                           '更新数据库
            rs.close
    End If
    HVrs.Close
    Set HVrs = nothing
End Sub
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charSet=gb2312">
<link rel="stylesheet" href="images/styles.css" type="text/css">
<title>聊天室</title>
</head>
<body leftmargin="0" Topmargin="0" marginwidth="0" marginheight="0">
<p align="center" style="margin-top: 10"><font size="6">聊天室注册向导</font></p>
<hr>
<center>
  <%
  If Request.QueryString("result")="successful" Then
        Call save()
        Call show()
  Else
        Call main()
  End If
 %>
</center>
<%Sub main()%>
<center>
<Form method="POST" action="register.asp?result=successful" name = registerform>
<center>
<table border="0" width="625" cellspacing="1" height="80">
 <tr>
      <td width="472" colspan="2" height="17" align = center>
        <p align="center"><b>基本信息设置</b>
        <p align="center">
    </td>
   </tr>
   <tr>
      <td width="126" height="8" bgcolor="#C0C0C0">
```

```
        <p align="center"><font color="#000000" size="2">用 户 帐 号:</font>
        </p>
    </td>
    <td width="485" height="8"><input name="UserName" size="32"   class="bk"> <font
            color="#FF0000"><font size="2">*</font></font><font color="#000000" size = "2">请
            输入您需要注册的用户名</font>
    </td>
</tr>
<tr>
    <td width="126" height="8" bgcolor="#C0C0C0">
        <p align="center">
        <font color="#000000" size="2">用 户 密 码:</font>
        </p>
    </td>
    <td width="485" height="1">
        <p align="left"><input type = "Password" name="UserPwd" size="32"
            class="bk"> <font color="#FF0000"><font size="2">*</font></font><font
            color="#000000" size = "2">用于登录系统的身份验证</font>
        </p>
</td>
</tr>
<tr>
    <td width="126" height="8" bgcolor="#C0C0C0">
        <p align="center">
        <font color="#000000" size="2">密 码 确 认:</font>
        </p>
    </td>
    <td width="485" height="25"><input type = "Password" name="Pwd" size="32"
        class="bk"> <font color="#FF0000"><font size="2">*</font></font><font
        size="2">确认密码</font>
    </td>
</tr>
<tr>
<td height="8" width="126" bgcolor="#C0C0C0">
    <p align="center"><font size="2">真 实 姓 名:</font></p>
</td>
<td height="1" width="559"><input type="text" name="Name" size="32"
        class="bk"> <font color="#FF0000"><font size="2">*</font></font><font
        size="2">请输入真实姓名</font></td>
</tr>
<tr>
<td height="8" width="126" bgcolor="#C0C0C0">
<p align="center"><font size="2">电 子 邮 箱:</font></p>
```

```
    </td>
    <td height="1" width="559"><input type="text" name="Email" size="32"  class="bk"
        style="ime-mode:disabled"> <font color="#FF0000"><font
        size="2">*</font></font><font size="2">请输入您的电子邮箱</font></td>
    </tr>
    <tr>
      <td width="472" colspan="2" height="21">
        <p align="center">
        <p align="center">
<input type="Submit" value="提交" name="register" class="Tips_bo">
    </td>
    </tr>
</table>
</Form>
<%End Sub%>
<%Sub show()%>
<table border="0" width="71%" cellspacing="1" height="47">
    <tr>
      <td width="100%" height="21">
      <p align="center"><font color="#FF0000"><b>用户注册成功</b></font>
      </td>
    </tr>
    <tr>
      <td width="100%" height="18">
        <p align="center"><font color="#FFffff">您的信息设置已经成功，进入聊天室请点击“完
            成”</font></p>
      <p align="center">[ <a href="index.asp"><b>完成</b></a>
      </td>
    </tr>
</table>
<%End Sub%>
</body>
</html>
<%
'关闭销毁对象
Set rs=nothing
conn.close
Set conn=nothing
%>
```

(2) 将创建的文件保存至 C:\Inetpub\wwwroot 文件夹中。

5. 创建聊天室主界面

聊天室主界面是一个多窗口的框架页面。该页面分为两部分，第一部分由 topmsg.asp 文件实现，显示聊天室名称；第二部分由 Chatroom.asp 文件实现，显示聊天室在线用户列表。Chatroom.asp 文件也是一个多窗口的页面，由 showname.asp、showmsg.asp 和 postmsg.asp 等 3 个文件组成。

【练习 13-29】创建聊天室名称显示页面 topmsg.asp。

(1) 创建网络聊天室名称显示文件 topmsg.asp，其代码如下所示：

```
<head>
<title></title>
</head>
<body topmargin="0" bgcolor="#C0C0C0">
<center>
<font size="7">
网络聊天室
</font>
</center>
<a href = logout.asp target = _parent>退出</a>
</body>
```

(2) 将 topmsg.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

【练习 13-30】创建聊天室在线用户列表文件 Chatroom.asp。

(1) 创建网络聊天室在线用户列表文件 Chatroom.asp，其代码如下所示：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>聊天室</title>
</head>
<frameset rows="*" cols="150,*" frameborder="1" border="1" framespacing="0">
  <frame border = 1 src="showname.asp?nmyself=<%=request.querystring("myself")%>"
      name="leftFrame" scrolling="yes" noresize>
  <frameset rows="*,100" frameborder="1" border="1" framespacing="0">
    <frame border = 1 src="showmsg.asp" name="showmsg"   scrolling = "yes">
    <frame border = 1
        src="postmsg.asp?myself=<%=request.querystring("myself")%>&toname=<%=request.Qu
        eryString("toname")%>" name="bottomFrame" scrolling="no" noresize>
  </frameset>
</frameset>
<noframes><body>
</body></noframes>
```

```
</html>
```

(2) 将 Chatroom.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

完成 topmsg.asp 文件和 Chatroom.asp 文件的创建工作后，用户可以参考下面练习所介绍的方法创建网络聊天室主界面框架文件 Main.asp 文件。

【练习 13-31】创建网络聊天室主界面框架文件 Main.asp。

(1) 创建网络聊天室主界面框架文件 Main.asp，其代码如下所示：

```
<html>
 <head>
  <title>聊天室</title>
 </head>
<!-- #include file="conn.asp" -->
<%
If Session("User")<>"" Then                           '用户登录
 '把用户名添加到在线名单中
 set rsx=Server.CreateObject("ADODB.RecordSet")        '创建 RecordSet 对象
 '从数据库中查询 id 为空的记录
 sqlx="select * from online where id is null"
 '执行查询，将结果保存在 rsx 中
 rsx.open sqlx,conn,1,3
 rsx.addnew                                            '添加一条新记录
 rsx("onlinetime")=time()                              '登录时间
 rsx("username")=Session("user")                       '用户名
 rsx.update                                            '更新数据库
 rsx.close                                             '关闭 RecordSet 对象
 Set rsx = nothing
%>
<frameset rows="15%,*" frameborder="1" border="1" framespacing="0">
  <frame src = "topmsg.asp" name = "topframe">
  <frame src="chatroom.asp?myself=<%=Session("user")%>" name="include">
  <noframes>
  <body>
</body>
  </noframes>
</frameset>
<frameset>
 </frameset>
</html>
<%
Else                                                   '用户没有登录
 '给出提示，返回到前一页
```

```
 Response.Write("<script>alert('你还没有登录!');history.back();</script>")
 '终止当前对 ASP 文件的执行，并将现有结果返回给客户端
 Response.End
End If
%>
```

(2) 将 Main.asp.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

6. 创建在线用户显示页面

下面将通过练习，介绍创建在线用户显示页面 ShowName.asp 的方法。

【练习 13-32】创建在线用户显示页面 ShowName.asp。

(1) 创建聊天室在线用户显示页面文件 ShowName.asp，其代码如下所示：

```
<html>
<head>
<title>用户列表</title>
<META HTTP-EQUIV="refresh" CONTENT="2">
</head>
<body>
<table height = "85%" valign = top>
<tr height = 10>
<td valign = top>
在线网友
<hr>
</td>
</tr>
<tr valign = top>
<td valign = top align = center>
<!-- #include file="conn.asp" -->
<%
'显示最新在线名单
set rsout=Server.CreateObject("ADODB.RecordSet")         '创建 RecordSet 对象
sqlout="select  * from  online  where id ORDER BY id DESC"
'从数据库中查询所有记录，按照 id 降序排列
rsout.open sqlout,conn,1,1                               '执行查询操作，将结果保存在 rsout 中
Do While not rsout.eof                                   '循环显示所有记录
    '建立超级链接
    response.write ("<a href = chatroom.asp?myself="&Session("user")&"&toname=")
    response.Write(rsout("username"))
    response.Write(" target = 'include'>")
    response.Write(rsout("username"))                    '显示用户名
    response.Write("</a>")
    response.Write("<br>")                               '换行
```

```
        rsout.movenext                                    '指向下一条记录
loop
response.Write("<br>")
%>
</td>
</tr>
<tr>
<td></td>
</tr>
</table>
<table border=0 cellpadding=0 cellspacing=0 width="86%" height="10%">
<tr>
<td></td>
</tr>
</table>
</body>
</html>
```

(2) 将 ShowName.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

7. 创建聊天发言提交页面

聊天发言提交页面是用户提交聊天发言的页面，用户可以参考下面练习所介绍的方法创建该页面。

【练习 13-33】创建聊天内容提交页面 PostMsg.asp。

(1) 创建聊天室中的聊天内容提交页面 PostMsg.asp，其代码如下所示：

```
<!-- #include file="conn.asp" -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>postmsg</title>
</head>
<body topmargin="0"   leftmargin="0">
<table width="100%" height="58" border="0">
<tr>
<td height="25" colspan="2">
<form action="postmsg.asp?act=addmsg&myself=<%=request.querystring("myself")%>
    &toname=<%=request.QueryString("toname")%>" method="post" name="form1" >
  <input readonly type = text name = FromName value = "<%=request.querystring("myself")%>">
  <select name="rname">
  <%If request.QueryString("toname") = "" Then%>
  <option>大家</option>
```

```
<%Else%>
<option><%=request.QueryString("toname")%></option>
<%
End If
'显示在线人数
set rsout=Server.CreateObject("ADODB.RecordSet")          '创建 RecordSet 对象
sqlout="select  * from  online  where id ORDER BY id DESC"
'从数据库中查询所有记录，按照 id 降序排列
rsout.open sqlout,conn,1,1                     '执行查询，将结果保存在 rsout 中
for i=1 to 50                                  '在下拉列表框中最多显示 50 个用户名
%>
    <option><%response.Write(rsout("username"))%></option>
    <%
    rsout.movenext                             '指向下一条记录
    if rsout.EOF Then Exit For                 '到达最后一条记录则跳出 For 循环
next
%>
</select>
说:
<input name="msg" type="text" size="30">
<input name="Submit" type="submit" value="提交发言" height="15">  <input name = "hidden"
    type = "submit" value="私聊">
<input name="username"  type="hidden" id="username"
    value="<%response.write(request.querystring("myself"))%>">
  </form></td>
</tr>
 <tr>
<td width="602" height="25"></td>
<td width="185">  </td>
</tr>
</table>
</body>
</html>
<%
 ac=request.QueryString("act")
'读取 act 的值
 if ac="addmsg" then
'act 的值为 addmsg
 If Request.Form("hidden")<>"" Then              '私聊
 Session("chat") = ""
 '清空 Session("chat")中的内容
 '把要显示的私聊信息保存在 Application("chat")中
  Application("chat") = "<font size = 2>"&request.Form("username")&"悄悄地对
```

```
                "&request.form("rname")&"说:<font color = black>"&request.form("msg")&"</font>
                "&time()&"</font>"
    Application("chatto") = request.form("rname")           '聊天对象
   Application("owner") = request.Form("username")          '发言者
                Else                                        '不是私聊
                set rs=server.createobject("adodb.recordset")  '创建 RecordSet 对象
                    sql="select * from chat"                '查询数据库
                    rs.open sql,conn,1,3                    '执行查询
                    rs.addnew                               '添加一条新记录
                    rs("FromName")=request.Form("username")         '发言者
                rs("ToName")=request.form("rname")          '聊天对象
                    rs("chat")=request.form("msg")          '显示私聊信息
                    rs("posttime")=time()                   '提交时间
                    rs.update                               '更新数据库
                    rs.Close                                '关闭 RecordSet 对象
                    set rs=nothing
                    '增加一条聊天记录，将 Application("Gchatnum")的值加 1
                Application("Gchatnum") = Application("Gchatnum")+1
            End If
                end if
%>
```

(2) 将 PostMsg.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

8. 创建聊天内容显示页面

聊天内容显示页面的效果由 ShowMsg.asp 文件实现，在该页面中可以显示聊天室中公共聊天的内容。

【练习 13-34】创建聊天内容显示页面 ShowMsg.asp。

(1) 创建聊天内容显示页面 ShowMsg.asp，其代码如下所示:

```
<!--#include file="conn.asp"-->
<HTML>
<HEAD>
<TITLE>聊天室</TITLE>
<META HTTP-EQUIV="refresh" CONTENT="2">
</HEAD>
<BODY>
<table width="100%" height="70%" border="0" >
  <tr>
    <td valign = top>
  <%
   '显示聊天内容
```

```
        'Session("Lchatnum") 为用户登录时聊天总记录数
        'Application("Gchatnum")为当前聊天总记录数
        'Session("Lchatnum")<Application("Gchatnum")表明用户登录后，该聊天室有用户聊天，
        '则显示登录之后的聊天记录
        If Session("Lchatnum")<Application("Gchatnum") Then
          start = Session("Lchatnum")+1
          for i = Application("Gchatnum") to start step -1    '使用 For 语句循环显示所有聊天记录
              set rs=Server.CreateObject("ADODB.RecordSet")       '创建 RecordSet 对象
              rs.Open "Select * from chat where id="&i,conn,1,3
              '从数据库中查询 id 值为 i 的记录，并将结果保存在 rs 中
      %>
          <a href = chatroom.asp?myself=<%=Session("user")%>&toname=<%=rs("FromName")%>
              target="include"><%response.Write(rs("FromName"))%></a>
          对
      <a href = chatroom.asp?myself=<%=Session("user")%>&toname=<%=rs("ToName")%>
target="include"><%response.Write(rs("ToName"))%></a>
      说:
      <font color = black>
      <%
      response.Write(rs("chat"))                    '聊天内容
      %>
      </font>
      <%
      response.Write(rs("posttime")&"<br>")         '发言时间
          %>
          <br>
        <%
              rs.close()                            '关闭 RecordSet 对象
              set rs=nothing
          Next
        End If
        %>
        </td>
      </tr>
    </table>
    <table height = "30%" width = "100%">
    <tr width = "10" valign = center height = "30%"   bgcolor = "#bbbbbb">
    <td><font color = red>私聊内容</font></td>
    </tr>
    <tr>
    <td valign = top>
    <%
    If Session("chat") <>Application("chat") Then     'Session("chat")用来保存用户浏览器上的私聊内容
```

```
'Application("chat")用来保存当前私聊内容，二者不相等，说明有新的私聊信息
'登录用户为私聊发言者或为私聊发言对象，或私聊对象为所有人，则显示私聊内容
 If Session("user") = Application("owner") or Session("user") = Application("chatto") or
          Application("chatto") = "大家" Then
        Response.Write Application("chat")                '显示私聊内容
        Session("chat") = Application("chat")             '保存用户可以看到的私聊内容
 End If
Else                                                      '不满足条件
 Response.Write Session("chat")                           '显示用户私聊内容
End If
%>
</td>
</tr>
</table>
</BODY>
</HTML>
```

(2) 将 ShowMsg.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

9. 创建聊天用户退出页面

下面将通过练习介绍用户退出聊天室文件 Logout.asp。

【练习 13-35】创建聊天室退出登录文件 Logout.asp。

(1) 创建聊天室退出登录文件 Logout.asp，其操作步骤如下所示：

```
<!--#include file = "conn.asp"-->
<%
 set rsdell=Server.CreateObject("ADODB.RecordSet")          '创建 RecordSet 对象
 '将当前登录用户从在线用户数据表中删除
 sqldell="delete * from online where username='"&Session("user")&"'"
 rsdell.open sqldell,conn,1,3                 '执行删除操作
 Session("user") = ""                         '清除 Session 中的用户名
 Session("chat") = ""                         '清除 Session 中的私聊内容
 Session("Lchatnum") = 0                      '将当前聊天记录数置为 0
 Response.Redirect "index.asp"                '页面转到登录界面
%>
```

(2) 将 Logout.asp 文件保存至 C:\Inetpub\wwwroot 文件夹中。

13.4.4　实例测试

完成本节实例的操作后启动 IE 浏览器，在浏览器地址栏输入 http://localhost/index.asp 并按 Enter 键显示聊天室登录页面，如图 13-65 所示。在聊天室登录页面中单击“注册新

用户”链接，可以打开如图 13-66 所示的聊天室新用户注册页面，注册新用户。



图 13-65　聊天室登录页面



图 13-66　聊天室用户注册页面

在图 13-64 所示的聊天室登录页面中成功登录聊天室后，将打开如图 13-67 所示的聊天室主界面，在该界面中，用户可以在页面的底部撰写并发送聊天内容，如图 13-68 所示。

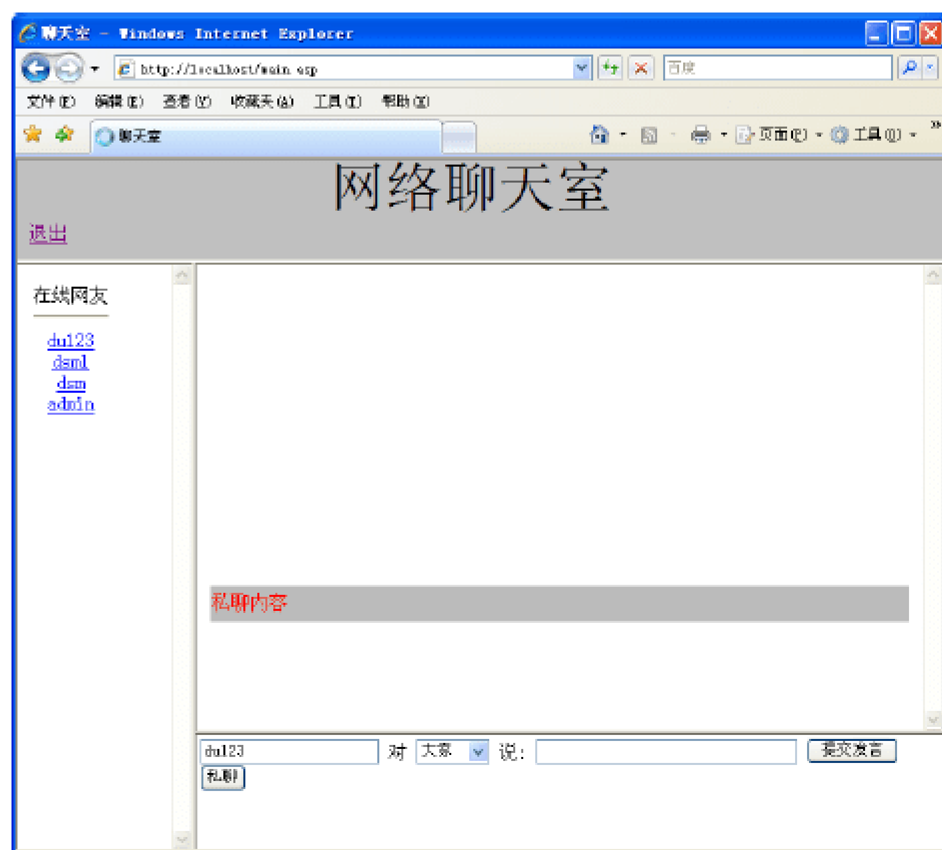


图 13-67　聊天室主界面



图 13-68　在聊天室中聊天

单击图 13-67 所示的聊天室主界面左上角的“退出”链接，可以退出聊天室模块。